Rectify Sensor Data in IoT: A Case Study on Enabling Process Mining for Logistic Process in an Air Cargo Terminal*

Preprint · August 2023

CITATIONS
0

READS
181

8 authors, including:

Chiao-Yun Li
RWTH Aachen University
11 PUBLICATIONS **48** CITATIONS
SEE PROFILE

Tejaswini Shinde
RWTH Aachen University
4 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Wil Van der Aalst
RWTH Aachen University
1,566 PUBLICATIONS **101,176** CITATIONS
SEE PROFILE

Rectify Sensor Data in IoT: A Case Study on Enabling Process Mining for Logistic Process in an Air Cargo Terminal*

Chiao-Yun Li[1,2], Aparna Joshi[1], Nicholas T. L. Tam[3], Sean Shing Fung Lau[3], Jinhui Huang[3], Tejaswini Shinde[1], and Wil M.P. van der Aalst[1]

[1] RWTH Aachen University, Aachen, Germany
{chiaoyun.li,wvdaalst}@pads.rwth-aachen.de
{aparna.joshi,tejaswini.shinde}@rwth-aachen.de
[2] Fraunhofer FIT, Birlinghoven Castle, Sankt Augustin, Germany
[3] Hong Kong Industrial Artificial Intelligence and Robotics Centre Limited, Shatin, NT, Hong Kong
{nicholastam,seanlau,gavinhuang}@hkflair.org

Abstract. The Internet of Things (IoT) has empowered enterprises to optimize process efficiency and productivity by analyzing sensor data. This can be achieved with process mining, a technology that enables organizations to extract valuable insights from data recorded during process execution, referred to as *event data* in a process mining context. In our case study, we aim to apply process mining to sensor data collected within a logistic process at an air cargo terminal, specifically from device-to-device communication. By representing the sensor data as event data, we rectify them to accurately capture the movement of package distribution in the logistic process. However, due to the communication dynamics, challenges arise from the presence of irrelevant data that does not impact the process instance's status. Moreover, issues such as faulty sensor readings and ambiguous data interpretation further compound these challenges. To overcome the obstacles, we collaborate with domain experts to develop rules that take into account the context of each event in a trace, enabling us to effectively capture package distribution within the system. We present the results of our process mining analysis, which have been validated by domain experts. This case study contributes to the understanding and utilization of sensor data for process mining in IoT environments, with a specific focus on data collected from device-to-device communication.

Keywords: Process mining · IoT · Sensor data · Logistic process · Data rectification · Device-to-device communication.

*This work was supported by the InnoHK funding launched by Innovation and Technology Commission, Hong Kong SAR. Additionally, we would like to thank Eric Poon, Bill Sio, and Sebastiaan van Zelst for their support.

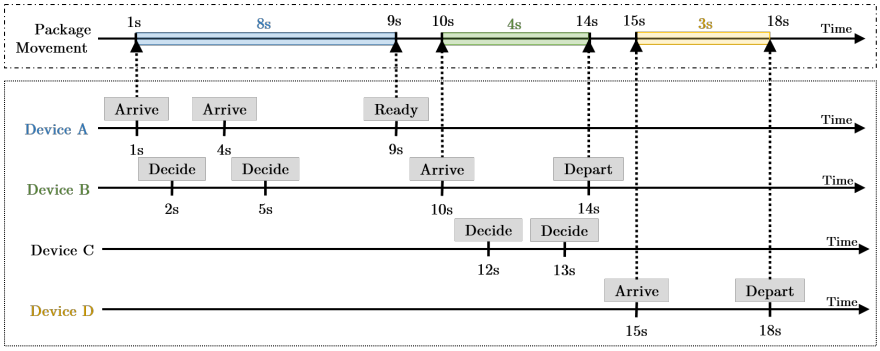
1 Introduction

The Internet of Things (IoT), a network of interconnected devices exchanging data through embedded sensors via the internet, has unlocked new possibilities for modern enterprises to digitize and automate their business processes [19]. Sensors integrated into devices collect valuable data about various aspects of a process such as machine conditions, location of an order, or individual health metrics. By analyzing these data, companies can derive actionable insights to improve efficiency and productivity in their processes [18].

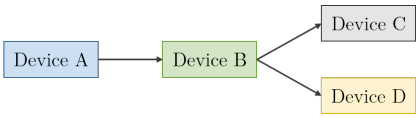
Process mining is a data-driven technology that empowers organizations to extract fact-based insights using *event data* generated and recorded in information systems during process execution [1]. For example, *process discovery* unveils the behavior of *activities* in a process [3, 5, 16], i.e., well-defined process steps, while *conformance checking* compares the observed behavior against predefined expectations [4,10,13]. By leveraging process mining on event data, organizations gain valuable insights for enhancing the execution and design of their processes.

Process mining typically assumes event data organized based on process instances (i.e., *cases*), with each event representing a status change within a case. In our case study of a logistic process in an air cargo terminal, our goal is to apply process mining to sensor data and extract insights on package distribution performance within the system. However, the sensor data collected primarily serves the purpose of device-to-device communication to determine device availability for the distribution process. When directly applying process mining to event data transformed from the sensor data, we may include events not directly relevant to package distribution, i.e., an event that does not signify a status change of package distribution within the system. Such misalignment between the sensor data and the event data hinders the application of process mining.

Figure 1 exemplifies the challenges described, necessitating the rectification of sensor data for process mining. The figure portrays the distribution of a package. The process beings with a package residing on device A, which communicates with device B to facilitate the package transfer. Following device A's readiness signal, the package is forwarded to device B. Subsequently, device B communicates with device C in an attempt to transfer the package; yet the latter declines to receive it. Consequently, the distribution proceeds to device D. Throughout the process, the sensor data contain extraneous data unrelated to the package's movement within the system. One example is the presence of data originating from device C, which the package never traversed. This example demonstrates how the specific sensor data contribute to the analysis of package distribution, which is the focus of this case study. In contrast, other data serve the purpose of communication between the involved devices and are considered extraneous noise that impedes the application of process mining. Such misalignment arises from the inherent divergence in their respective purposes. The sensor data utilized in our case study is primarily intended to support and enable device-to-device communication, which stands as one of the fundamental objectives of sensors within an IoT environment. Hence, to effectively apply process mining, it is crucial to rectify and align the sensor data with the behavior of interest.



(a) Misalignment between the sensor data and the event data indicating the package movement in the system. The lower timelines depict data samples collected by sensors on specific devices for a package distribution at a given point in time, labeled with the signals (e.g., Arrived, Departed) that indicate the availability of the corresponding device. The upper timeline displays the actual package movement. Dashed arrows highlight the data samples that *actually* signify the package movement.



(b) Relationship of devices in the system, with arrows indicating the possible direction of package distribution on the devices.

Fig. 1: An example of sensor data for device-to-device communication in the context of a package distribution. Consistent coloring is utilized to represent information related to each device across the relationship of devices, the timelines of signals sent by devices, and the timeline representing the package's stay on each device.

In this paper, we illustrate the rectification process applied to the collected sensor data in our case study. Through careful analysis, we identified and examined the challenges arising from the misalignment between the communication among the involved devices and the *actual* distribution of packages. Practical challenges, such as legacy systems and the ambiguity in interpreting sensor data (e.g., the departure of a package from a device may be signified by various signals like readiness or departed, as exemplified in Figure 1), further contribute to the complexity of the rectification process. We overcome these challenges and rectify the sensor data to align them with the package distribution within the terminal. This collaborative effort involves leveraging the expertise of domain professionals to ensure the reliability of the decisions made in the solutions. The effectiveness of our approach is demonstrated through an analysis of the behavior of package distribution using the rectified event data with process mining techniques, leading to valuable and validated insights.

Table 1: Every row represents a data sample from a sensor installed on a conveyor device (DID) to exchange its status (Sig) in relation to a package (PID), which is placed in a tray (TID) to be distributed at a specific point in time (Timestamp). The device's location is identified by its associated floor and zone, and the type of device (Type), such as conveyor belt or lift shaft, is also provided.

TID	PID	Sig	DID	Timestamp	Floor	Type	Zone	...
FFD541256	2365884459	Initiate	KJDQ4414	12:05:23	9	LB	J	...
FFD541256	2365884459	Ready	KJDQ4414	12:05:24	9	LB	J	...
FFD541256	2365884459	Ready	KJDQ4414	12:14:30	9	LB	J	...
FFD541256	2365884459	Decide	BRMI1121	12:15:35	0	RS	J	...
FFD541256	2365884459	Arrive	BRMI1121	12:16:31	0	RS	J	...
FFD541256	2365884459	Depart	BRMI1121	12:17:25	0	RS	J	...
FFD541256	2365884459	Arrive	UGOI9833	12:17:26	2	CB	J	...
FFD541256	2365884459	Arrive	NXVR3307	12:17:35	2	CB	J	...

The remainder of the paper is structured as follows. In Section 2, we introduce the available sensor data and provide an example of rectification. Section 3 illustrates the challenges identified and the approach developed. In Section 4, we apply process mining on the repaired sensor data and demonstrate the outcomes. Section 5 presents related work in the field, while Section 6 concludes the paper by summarizing the lessons learned from our case study.

2 Overview

In this section, we present a sample of sensor data provided and an example illustrating the rectification.[1]

Representation of Sensor Data as Event Data. Table 1 presents an excerpt of the sensor data collected. Each row corresponds to a data sample generated by a sensor for communication between the devices. For example, the first row specifies that the package 2365884459 is in the tray FFD541256 on device KJDQ4414 with type LB, which *initiates* the package distribution in zone J (written as ZJ) on floor 9 (written as F9) at 12:05:23.

We consider a data sample an event. A case consists of events describing package distribution, identified by the package identifier PID. Except for TID, which is a case attribute, other fields are assigned as event attributes. A *trace* is a sequence of events in a case ordered based on their timestamps as visualized in Figure 2, and an *event log* is the collection of traces.

[1] Due to confidentiality, the data are manipulated and anonymized, while preserving the relative relationships between data samples to illustrate the observed behavior in the paper.

Fig. 2: A visualization of a fragment of the trace based on Table 1, where every chevron represents an event annotated with the corresponding attributes, i.e., timestamp and signal, and is colored based on its *activity*, i.e., device identifier.

Rectification of Event Data for Package Movement. Our objective is to align event data with the package movement throughout its distribution. We aim to determine the specific device on which the package resides and the corresponding timeframe. Figure 3 presents an excerpt of the rectification. The package `2365884459` was on `KJDQ4414` from 12:05:23 to 12:14:30, which is inferred by the first and last events among the continuous events of `KJDQ4414`. Next, the package arrived and departed `BRMI1121` at 12:16:31 and 12:17:25, respectively, as indicated by the events of the arrival and departure of the package on `BRMI1121`. Then, the package arrived `UGOI9833` at 12:17:26; without another event for the package on `UGOI9833`, we assume that the departure occurred at the same time as it arrived at the next device, i.e., 12:17:35. In this example, we demonstrate some rectification performed, involving renaming the signals, filtering out communication overhead, and creating an artificial event.

We present a sample of the provided sensor data and illustrate the rectification conducted through an example. By showcasing the raw sensor data and its corresponding repair, this section highlights the necessity of rectifying sensor data within the context of this case study.

3 Event Data Rectification

We identified several challenges in aligning the event data with the package movement. To address the challenges, we drew on domain knowledge and developed a rectification process to repair the event data.

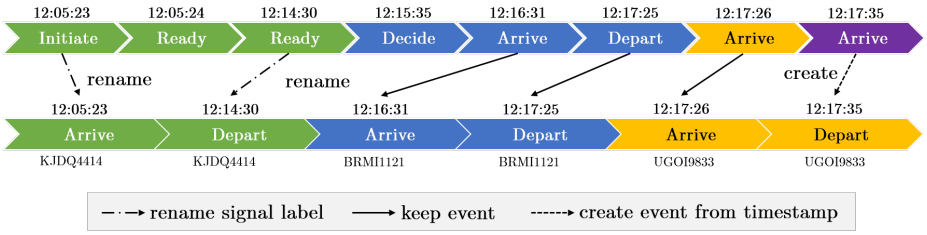


Fig. 3: Rectifying event data based on the sensor data presented in Table 1. The top figure depicts the trace fragment in Figure 2. The bottom one illustrates the rectified trace fragment that aligns with the package movement in the system. Different arrow types are used to distinguish the implemented solutions.

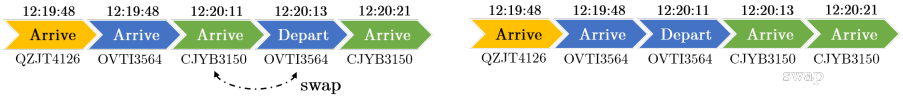
3.1 Challenges

We analyzed the event data and discovered some challenges specific to sensors and the system. In the IoT environment, device-to-device communication facilitates information exchange. However, this communication dynamic also introduces a challenge where some events derived from the transformed event data are not directly related to the physical movement of a package. For instance, the events labeled with `Decide` in Figure 1 and Table 1 are specifically transmitted to assess the suitability of the respective device for package reception. Another example is depicted by the second `Arrive` from device A in Figure 1, which is regarded as timeout noise resulting from an unexpectedly extended duration of stay on the device.

Furthermore, we discovered an improbable situation depicted in Figure 4a, where a package appeared to be simultaneously present on two devices, as indicated by its arrival at `CJYB3150` before its departure from `OVTI3564`. Collaborating with domain experts, subsequent investigation revealed that this unrealistic behavior was due to faulty sensors – `CJYB3150` detects the arrival of the package before `OVTI3564` signifies its departure. To address this issue, we implemented a solution by interchanging the timestamps, as illustrated in Figure 4b.

Finally, the devices in our case study exhibit varying communication patterns, where not all devices are capable of sending all types of messages. This diversity in programming logic among the devices results in different combinations of signals being transmitted, as illustrated in Figure 2. Additionally, when a package distribution encounters obstacles and a device along its path is not ready for package dispatch or reception, further communication is required. However, the communication pattern may not be universally applicable to other devices in the system. As a result, the interpretation of a signal extends beyond its literal meaning and relies on the contextual information associated with the corresponding event. For instance, in Figure 3, the second `Ready` is aligned to the departure from `KJDQ4414` since no events are labeled as `Depart` from `KJDQ4414` before the package arrives on `BRMI1121`.

The challenges encountered in our case study have implications that extend beyond our specific scenario and are relevant to various IoT settings involving sensor data and physical object movement. First, the presence of extraneous data



(a) Event data with sensor fault, where a package arrives on the next device before departing the previous device.

(b) Repaired event data with sensor fault, where the timestamps of the implausible arrival and departure are swapped.

Fig. 4: Repairing event data with sensor fault. We highlight the corrective measures implemented on the observed behavior in Figure 4a, resulting in the behavior depicted in Figure 4b.

resulting from device-to-device communication dynamics can introduce noise in process mining. For instance, an online order or a package delivery experiences repeated notifications of *waiting* or delayed status for several days, despite the absence of any meaningful progress or updates from the business perspective. Second, the presence of faulty sensors is not limited to our case study. In different contexts, such as logistics or manufacturing, faulty sensors can produce misleading readings, causing a discrepancy between the perceived status of a case and its actual condition. Finally, the existence of different programming logic among sensors introduces additional challenges in process mining. This challenge is not limited to legacy systems, as demonstrated in our case study, but also extends to IoT environments encompassing devices from different manufacturers. These challenges emphasize the importance of rectifying sensor data in IoT environments to ensure the reliability of the insights obtained through process mining.

3.2 Rectification Process

We develop a rectification process based on three principles identified during the analysis. This section outlines the process and we further illustrate the principles.

Overview. With the aim of reducing ambiguities, we developed a rectification process outlined in Figure 5. The process consists of three phases. First, we process the event data using explicit business rules to handle the data quality issues arising from the data extraction process. Second, we address ambiguities and filter out noise, which includes events that are not directly associated with the physical movement of a package, as well as events that exhibit improbable behavior resulting from sensor malfunctions. Finally, we merge the consecutive events from the same device and relabel the events for the arrival and departure of a package on each device.

Principles. We demonstrate the principles that we applied for the rectification of the event data in the process.

1. **Signal category.** There are eight different signals. We categorize them based on their literal meaning, which is described in Table 2. The first category, *physical movement*, includes signals that primarily indicate the physical

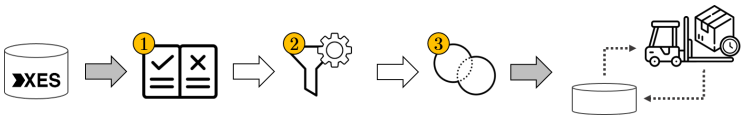


Fig. 5: Schematic diagram of the rectification process. First, we process the event data based on explicit rules. Next, we filter the events solely for communication messages and noises. Then, we merge and relabel events to indicate the stay of a package on a device. The resulting rectified event data signify the package movements in reality.

Table 2: Categories of signals based on their literal meaning and domain knowledge.

Category	Description	Signal Values (`Sig`)
Physical Movement	*Mostly* indicating the physical movement of a package	`Arrive`, `Depart`
Communication	*Often* for exchanging the status of the device	`Decide`, `Ready`, `Initiate`
Package Distribution	*Always* updating the distribution of a package	`Cancel`, `Insert`, `Change`

movement of a package. Next, the category of *communication* consists of signals that often show the status of a device. Lastly, the category of *package distribution* comprises signals that indicate the distribution of a package, e.g., a package distribution is canceled with an event labeled with `Cancel`.

2. **Certainty based on signals.** Due to the challenges identified, the categories defined are insufficient to determine the actual arrival or departure of a package on a device Hence, in collaboration with domain experts, we establish a ranking of the *certainty*, i.e., whether an event indicates the *actual* movement of a package, based on its signal, taking into account the context of the event in a trace. This ranking is separately defined for the arrival and departure of a package, allowing us to demonstrate the relative certainty associated with each event in relation to its context.
   - *Certainty for arrival:* `Arrive`, `Initiate`, `Decide`, `Ready`
   - *Certainty for departure:* `Depart`, `Ready`, `Decide`, `Initiate`

   Note that an `Arrive` only signifies the arrival of a package on a device, and a `Depart` indicates the departure; hence, they are not considered in the complementary ranking.
3. **Interpretation based on the context.** In addition to the literal meaning of the signals, whether an event indicates actual package movement depends on the context of the event. For instance, in the case of the second `Ready` from `KJDQ4414` in Figure 3, it is considered as indicating actual movement because there are no events labeled as `Depart` from `KJDQ4414`.

Due to the large number of devices involved, it is impractical to identify and resolve all the ambiguities with the assistance of domain experts. Moreover, as the system was constructed long ago and some of the original domain experts are no longer associated with the organization, the available domain knowledge for addressing these ambiguities is limited. Reprogramming all the sensors solely for the purpose of process mining is not a viable option due to time and budget constraints. Hence, we define these principles to guide the rectification of various solutions in each phase of the process, incorporating limited domain knowledge and effectively addressing the ambiguities.

3.3 Application of Principles

In this section, we present examples that demonstrate the application of the principles in each phase of the rectification process. The implementation of the

solutions follows a rule-based and automated approach, which has been developed through rigorous testing and iterative refinement. We carefully define the rules through extensive analysis of the event data, along with discussions and validation with domain experts. Our objective is to minimize the potential for false corrections during the automated rectification process. This iterative approach allowed us to continually enhance the effectiveness and accuracy of the automated solutions, ensuring the reliable of the results.

For simplicity, we adopt the same expression to represent a trace fragment, without displaying the timestamp or device identifier. First, we address the data quality issues arising from the collection and extraction of sensor data.

Example 1: Identify incomplete cases. The presence of `Arrive` is assigned with significantly higher importance compared to other signals. A package is considered more likely to actually reach a device when an `Arrive` is sent from the device. By prioritizing `Arrive`, we address the completeness of cases by considering those without an `Arrive` event as incomplete.

Example 2: Reorder events with identical timestamps. We observed a peculiar behavior in which a package appears to be rapidly shuttled back and forth between two pieces of devices within an unreasonably short duration (less than a second). This behavior is impossible within the normal operation of the system. The anomaly may stem from the arrival of data samples at the data lake in an order that does not correspond to the package movement. We address it by reordering such events based on their context as shown in Figure 6.

Next, we resolve the ambiguities arising from the challenges discussed in Section 3.1 and filter out noises. The following examples illustrate the solution implemented to address four types of noise caused by ineffective cancellations, timeouts, communication, and sensor faults.

Example 3: Detect effective cancellations. Cancellation of package distribution occurs due to business reasons. The cancellation can be reversed by an `Insert`. When a cancellation is retracted, we consider the associated events as noise. However, the relationship between a `Cancel` and an `Insert` is undefined. To establish their relation, we determine their proximity in a trace. Retraction is considered effective within a *distance* of 2 in the trace as depicted in Figure 7.

(a) Swap events based on surrounding device identifiers.

(b) Reorder events while preserving the order for the other events.

Fig. 6: Reorder events of identical timestamps based on the context. Events sharing the same timestamp are grouped and highlighted, with additional emphasis on the correction focus.

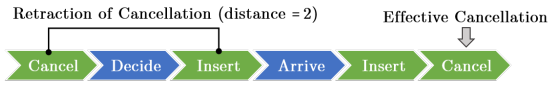


Fig. 7: Detection of retraction of cancellation, noise (the second `Insert`), and effective cancellation (the last `Cancel` since there is no subsequent `Insert`).

Example 4: Remove timeout noise. Events labeled with the same signal from the same device are classified as timeout noise and are eliminated based on the defined ranking. For instance, in the case of `Arrive`, the timeout noise refers to those that are not the first occurrence among consecutive events from the same device. The timeout noise for `Depart` is defined symmetrically. Figure 8 highlights the events as timeout noise under other different conditions.

(a) First `Ready` is timeout noise due to `Arrive` from the same device.

(b) Last `Ready` is timeout noise due to `Depart` from the same device.

(c) Last `Initiate` is timeout noise based on `Ready` in the certainty ranking.

Fig. 8: Identification of timeout noise across various scenarios. We identify timeout noise based on the context and the defined certainty ranking, which are highlighted with arrows.

Example 5: Filter communication noises. According to the ranking of certainty, we identify communication noise based on context. If an event is surrounded by other events with signals that rank higher according to the defined ranking, we classify it as communication overhead and exclude it from the event data. The examples in Figure 9 demonstrate the mechanism and the identification of communication noise across various scenarios.

(a) Both `Decide` and `Ready` are considered as communication overhead due to the existence of both `Arrive` and `Depart`.

(b) `Decide` is regarded as communication overhead while `Ready` is not due to the presence of `Arrive` and the absence of `Depart`.

Fig. 9: Recognition of communication noise based on category and certainty ranking. We point out the communication overhead with arrows.

Example 6: Swap timestamps for faulty sensors. Based on the high confidence placed on `Arrive` for the actual arrival, we identify noise caused by malfunctioning sensors and address it by swapping the timestamps, as explained in Figure 4.

This phase of the process heavily relies on the context to identify and eliminate noise. As the context evolves, we iteratively apply the solutions to reduce the communication overhead. This process continues until no further events can be removed. Once the communication overhead is eliminated, in the next phase, we relabel and create artificial events to align them with the package movement and ensure the consistency of event format per device.

Example 7: Relabel events. Suppose only two events of a device remain that exhibit a clear logical order in a trace; relabeling is not required or is straightforward. In situations where only one `Arrive` is sent, we create an artificial event to represent the departure, using the timestamp of the next device's arrival in the package distribution. If only a `Depart` exists, the implementation follows a symmetrical approach. Figure 3 demonstrates the scenarios described. Note that these decisions are based on context and category. If only communication signals exist, we assume the package never reaches the device and consequently remove the associated events. For example, in Figure 1, the events from device C are appropriately eliminated during this step due to the absence of signals in the category of physical movement.

Building upon the principles, we have effectively devised solutions to tackle the challenges discussed in Section 3.1. These solutions have been seamlessly integrated into our project partner's information system, enabling the computation of proprietary key performance indicators. By leveraging the rectified event data extracted through our proposed solutions, we have enabled empowered analysis and facilitated process mining activities, as presented in the next section.

4 Process Mining

The rectified event data consists of approximately 5,000 unique device identifiers and 20,000 distinct variants. Given the complexity of the event data, validating every path with the physical situation in the terminal is not feasible. Meanwhile, the classical discovery and conformance-checking algorithms fall short in terms of scalability. Hence, we abstract the event data based on the device attributes to uncover process models and validate the insights obtained from process mining outcomes instead. The models are discovered through inductive mining techniques and further enhanced with domain knowledge [11, 12], which are colored with the relative frequency of the distribution and annotated with labels for readability. The abstraction is performed as in Figure 10, where we *merge* and rename the events based on `floor`. Abstraction based on other attributes is performed in a similar manner.

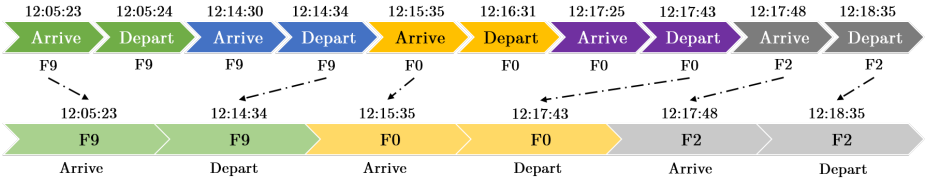


Fig. 10: Abstracting event data based on `floor` attribute. The top trace showcases a fragment of rectified trace and the bottom one is the corresponding abstracted fragment based on *floor*.

Figure 11 and 12 present the models discovered based on the location information. Figure 11 reveals that most of the package distributions are initiated on `F9` but can end up on different floors, mainly exiting on `F0`. Except for `F0`, most floors are rarely revisited. It is also worth noting that not all the data samples from the sensors on lift shaft (`LS`) are received when packages are distributed across floors, indicating missing sensor readings. Figure 12 demonstrates the distribution based on zones in `F0`. Two stages are identified. In the first stage, no dominant paths across zones are identified; meanwhile, some zones are closely related based on their values, which reflects the geographical naming conventions. In the second stage, the distributions leave the `F0` from zones `U`, `K` and `R`.

Figure 13 presents another aspect of package distribution. Similarly, we see the combination of device types that are often applied together. Moreover, it shows a sequential pattern across the 3 stages of distribution: the beginning, during, and end on `F0`. Besides packages arriving from lift shafts (`LS`), some packages are stacked in storage-type devices, i.e., hand-operated lift (`HL`) and lifting boom (`LB`), before being distributed throughout the floor.

We evaluate the fitness [2] and the precision [15] of the models based on the event data before and after rectification. At the floor level, the fitness is approximately 0.7 for both datasets, while the precision is 0.97 and 0.93 for the pre- and post-rectification datasets, respectively. We assume that the models in Figure 12 and 13 represent behavior on all floors and compare them against the package distribution on other floors. Figure 14 presents the results. The metrics do not differ much at the floor and zone levels. However, regarding device type, the fitness increases for most floors, while the change in precision

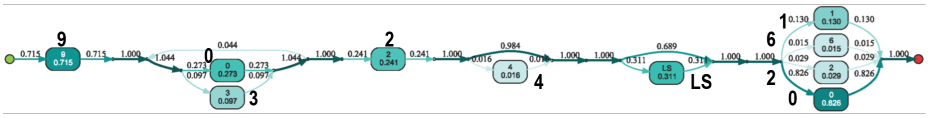


Fig. 11: Package distribution based on floors. Since lift shafts (device of type `LS`) is used for distributing packages across different floors, we do not classify them to any floors.

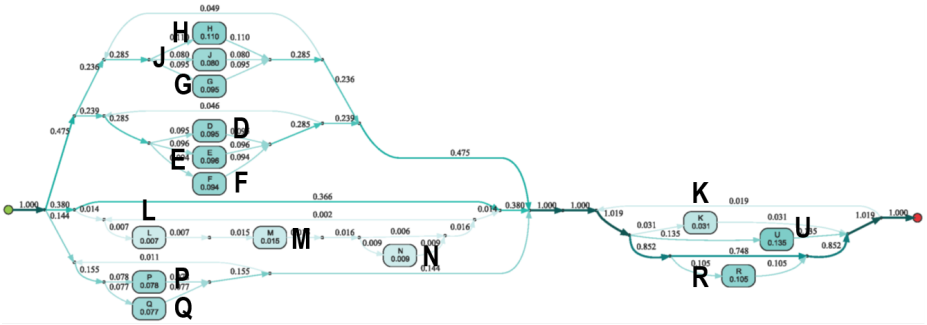


Fig. 12: Package distribution based on zones on `F0`.

varies depending on the floor. In most cases, F-Score is enhanced. Moreover, the metrics for `F9` exhibit lower fitness values due to the limited device types on the floor, which primarily serves as the entry floor for package distribution.

5 Related Work

Sensor data in the context of the IoT present unique challenges in terms of quality and reliability. Extensive research has been conducted to address these challenges. Teh et al. conducted a systematic review focusing on the quality-related issues of physical sensor data, categorizing eight types of sensor data errors and discussing existing solutions for error detection and correction [17]. Similarly, Gaddam et al. provided a comprehensive review that specifically examined the detection of sensor faults in the IoT [8]. Additionally, Mansouri et al. identified and discussed various IoT data quality issues based on existing research [14]. These issues align closely with the challenges encountered in our case study, including the misalignment arising from faulty sensors, inconsistencies due to different sensor programming logic, redundancy owing to device-to-device communication, and ambiguity in data interpretation. However, while these existing techniques aim to tackle general data quality issues in IoT, they may not directly address the specific challenges encountered in our case study, which focuses on analyzing and extracting insights from package distribution within the system.

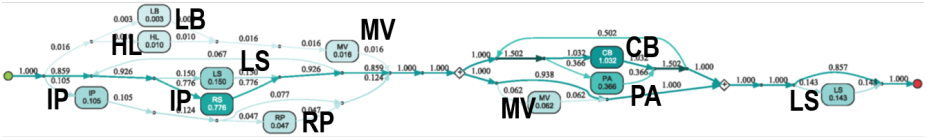
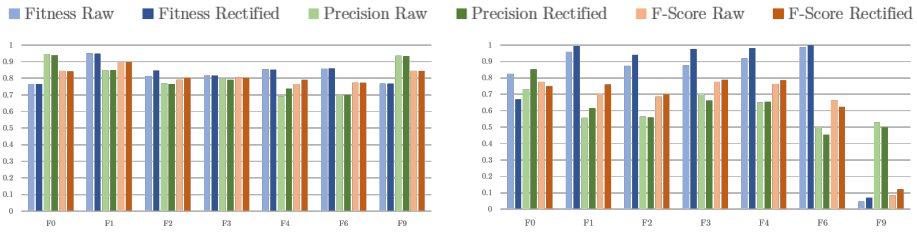


Fig. 13: Discovering package distribution on `F0` at the abstraction level of device type.



(a) Metrics based on the model in Figure 12.

(b) Metrics based on the model in Figure 13.

Fig. 14: The fitness, precision, and F-score of the models based on `F0`, which are compared against two datasets per floor: *raw* (before rectification) and *rectified*. The metrics are color-coded consistently, with the hue differentiating datasets.

Our approach rectifies sensor data by considering the real-world behavior of package distribution and addresses the ambiguity in data interpretation through the context of an event in a case. The customized approach bridges the gap between the data used for process mining and the specific data quality considerations of our case study. These findings highlight the relevance and the potential applicability of our work in addressing the broader challenges in IoT.

Process mining has emerged as a prominent technology in the IoT domain, enabling the analysis of sensor data collected in IoT environments. For instance, Dreher et al. explored the feasibility and application of process mining in manufacturing-related processes [6]. Considering the similarities between logistic processes and manufacturing processes, where the efficient flow of goods and services is a crucial objective, the research gap identified in manufacturing-related processes is also relevant to our case study. Specifically, the paper acknowledges the research gap we aim to address in this case study, stating that "implementing process mining in manufacturing faces a significant disconnect between the physical flow of materials and the digital information flow" [6]. Janssen proposed a technique to discretize sensor data into event data suitable for process mining by correlating events, discovering activities, and abstracting events [9]. The work focuses on elevating the sensor data to the business level. Similarly, van Eck et al. conducted a study where they abstracted sensor data by mapping temperature and acceleration measurements from a smart baby bottle to human activities and identifying process instances through activity grouping [7]. By applying process mining to the transformed data, their work demonstrated the value of process mining in facilitating the design process of smart products. Our case study shares the same objective, aiming to identify significant events or sensor data that represent meaningful status changes within a case, facilitating analysis at the business level. However, our study specifically addresses the challenges posed by device-to-device communication and focuses on resolving the ambiguity in data interpretation based on contextual information.

In conclusion, our case study addresses the specific challenges that arise from device-to-device communication in the context of process mining. While existing techniques for addressing sensor data issues in IoT environments may not directly apply to our case study, we have developed customized rules tailored to the challenges observed in the real-world package distribution scenario. This tailored approach benefits from the iterative discussions and presentations with stakeholders and domain experts, ensuring the effectiveness and reliability of our approach. Moreover, existing research in process mining predominantly focuses on identifying key concepts such as activities and cases from sensor data capturing continuous measurements. In contrast, our case study fills an important gap by proposing an approach to address the integration of sensor data from device-to-device communication within the process mining framework.

6 Lesson Learned and Opportunities

Sensor data present unique challenges in applying process mining to extract business-level insights. In addition to the high volume of data points typically found in sensor data, we identified various inherent challenges including sensor malfunctions, missing readings, and communication overhead in the case study. Furthermore, the limited availability of domain knowledge in a legacy system adds to the complexity, with uncertainties arising in the interpretation of sensor data. Meanwhile, conducting a simulation on a large system is expensive. To tackle these challenges, we developed a rectification process based on the principles identified and discussed throughout our analysis. The solutions were implemented specifically tailored to the identified conditions, to effectively repair sensor data and align them with the package movement. We demonstrate the effectiveness of the proposed solutions with the validated process mining outcomes based on the rectified event data. For future work, although the implemented solutions have been customized to achieve optimal quality, there is an opportunity to apply them to a broader range of IoT use cases. By utilizing the certainty ranking and conducting repeated checks on the contextual information of events, a general solution is to be developed to *match* event pairs for every package distribution on a device to enhance the applicability and effectiveness of our approach.

References

1. van der Aalst, W.M.P.: Process Mining - Data Science in Action, Second Edition. Springer (2016)
2. van der Aalst, W.M.P., Adriansyah, A., van Dongen, B.F.: Replaying history on process models for conformance checking and performance analysis. WIREs Data Mining Knowl. Discov. **2**(2), 182–192 (2012)
3. Banziger, R., Basukoski, A., Chaussalet, T.J.: Discovering business processes in CRM systems by leveraging unstructured text data. In: HPCC/SmartCity/DSS. pp. 1571–1577. IEEE (2018)

4. Bauer, M., van der Aa, H., Weidlich, M.: Sampling and approximation techniques for efficient process conformance checking. Inf. Syst. **104**, 101666 (2022)
5. Chiudinelli, L., Dagliati, A., Tibollo, V., Albasini, S., Geifman, N., Peek, N., Holmes, J.H., Corsi, F., Bellazzi, R., Sacchi, L.: Mining post-surgical care processes in breast cancer patients. Artif. Intell. Medicine **105**, 101855 (2020)
6. Dreher, S., Reimann, P., Gröger, C.: Application fields and research gaps of process mining in manufacturing companies. In: Reussner, R.H., Koziolek, A., Heinrich, R. (eds.) 50. Jahrestagung der Gesellschaft für Informatik, INFORMATIK 2020 - Back to the Future, Karlsruhe, Germany, 28. September - 2. Oktober 2020. LNI, vol. P-307, pp. 621–634. GI (2020)
7. van Eck, M.L., Sidorova, N., van der Aalst, W.M.P.: Enabling process mining on sensor data from smart products. In: Tenth IEEE International Conference on Research Challenges in Information Science, RCIS 2016, Grenoble, France, June 1-3, 2016. pp. 1–12. IEEE (2016)
8. Gaddam, A., Wilkin, T., Angelova, M., Gaddam, J.: Detecting sensor faults, anomalies and outliers in the internet of things: A survey on the challenges and solutions. Electronics **9**(3), 511 (2020)
9. Janssen, D., Mannhardt, F., Koschmider, A., van Zelst, S.: Process model discovery from sensor event data. In: Leemans, S.J.J., Leopold, H. (eds.) Process Mining Workshops - ICPM 2020 International Workshops, Padua, Italy, October 5-8, 2020, Revised Selected Papers. Lecture Notes in Business Information Processing, vol. 406, pp. 69–81. Springer (2020)
10. Leemans, S.J.J., van der Aalst, W.M.P., Brockhoff, T., Polyvyanyy, A.: Stochastic process mining: Earth movers' stochastic conformance. Inf. Syst. **102**, 101724 (2021)
11. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Using life cycle information in process discovery. In: Reichert, M., Reijers, H.A. (eds.) Business Process Management Workshops - BPM 2015, 13th International Workshops, Innsbruck, Austria, August 31 - September 3, 2015, Revised Papers. Lecture Notes in Business Information Processing, vol. 256, pp. 204–217. Springer (2015)
12. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Scalable process discovery and conformance checking. Softw. Syst. Model. **17**(2), 599–631 (2018)
13. de Leoni, M., Munoz-Gama, J., Carmona, J., van der Aalst, W.M.P.: Decomposing alignment-based conformance checking of data-aware process models. In: Meersman, R., Panetto, H., Dillon, T.S., Missikoff, M., Liu, L., Pastor, O., Cuzzocrea, A., Sellis, T.K. (eds.) On the Move to Meaningful Internet Systems: OTM 2014 Conferences - Confederated International Conferences: CoopIS, and ODBASE 2014, Amantea, Italy, October 27-31, 2014, Proceedings. Lecture Notes in Computer Science, vol. 8841, pp. 3–20. Springer (2014)
14. Mansouri, T., Moghadam, M.R.S., Monshizadeh, F., Zareravasan, A.: Iot data quality issues and potential solutions: A literature review. Comput. J. **66**(3), 615–625 (2023)
15. Munoz-Gama, J., Carmona, J.: A fresh look at precision in process conformance. In: BPM. Lecture Notes in Computer Science, vol. 6336, pp. 211–226. Springer (2010)
16. Pan, Y., Zhang, L.: Automated process discovery from event logs in bim construction projects. Automation in Construction **127**, 103713 (2021)
17. Teh, H.Y., Kempa-Liehr, A.W., Wang, K.I.: Sensor data quality: a systematic review. J. Big Data **7**(1), 11 (2020)

18. Valencia-Parra, Á., Ramos-Gutiérrez, B., Varela-Vaca, A.J., López, M.T.G., Bernal, A.G.: Enabling process mining in aircraft manufactures: extracting event logs and discovering processes from complex data. In: BPM (Industry Forum). CEUR Workshop Proceedings, vol. 2428, pp. 166–177. CEUR-WS.org (2019)
19. Wójcicki, K., Biegańska, M., Paliwoda, B., Górna, J.: Internet of things in industry: Research profiling, application, challenges and opportunities—a review. Energies **15**(5), 1806 (2022)