Leveraging Business Process Mining to Obtain Business Intelligence and Improve Organizational Performance

Inaugural Dissertation to Obtain the Academic Degree of a Doctor in Business Administration at the University of Mannheim

submitted by

Fareed Zandkarimi, Master of Science
Mannheim

Dean:

Joachim Lutz

Referent:

Prof. Dr. Hartmut Hoehle

Chair of Enterprise Systems

University of Mannheim

Correferent:

Prof. Dr. Christoph Bode

Endowed Chair of Procurement

University of Mannheim

Day of oral examination: 14.12.2023

Contents

List of Figures

List of Tables

1 Introduction

> *"If you cannot describe what you are doing as a process, you do not know what you are doing."* (Deming, 2018)

Defined as a series of interconnected activities and tasks that are designed to achieve specific business objectives, business processes play a significant role in ensuring operational efficiency and effectiveness (Dumas et al., 2018). Business processes encompass the flow of information, materials, and activities within an organization, and guide the efficient and effective execution of tasks. In essence, business processes serve as the foundation for how work is organized, coordinated, and performed in enterprises (Davenport et al., 1990; Harmon, 2019; Van Looy & Shafagatova, 2016).

Studying business processes is of utmost importance for several reasons. First, gaining a thorough understanding of existing processes allows organizations to identify areas that can be optimized, leading to improved performance and cost-effectiveness (Elbashir et al., 2008; Gebauer & Schober, 2006). Second, by studying business processes, organizations can detect bottlenecks or areas of inefficiency and ineffectiveness, enabling targeted process improvement efforts (Melville et al., 2004). Third, analyzing business processes provides valuable insights into how different parts of the organization are interconnected thereby fostering better coordination and alignment across departments (McCormack & Johnson, 2001). Lastly, studying business processes is essential for successfully implementing enterprise systems, as it ensures that these systems are tailored to meet the specific needs and workflows of the organization (Dumas et al., 2018).

Analyzing the execution of processes in real-world scenarios provides valuable insights into their actual performance, enabling organizations to optimize their workflows and enhance overall productivity (Sakr et al., 2018). In fact, the evaluation of digital footprints of processes assumes critical importance due to their proximity to the actual performance of organizations, bringing more transparency to the underlying workflows. The availability of abundant event data further amplifies the significance of process analysis, creating new opportunities for in-depth investigation and improvement (Van der Aalst, 2016).

In this context, Process-Aware Information Systems (PAIS) emerge as crucial components, having a long-standing presence in various domains for decades (Dumas et al., 2005). Enterprise Systems, including widely adopted Enterprise Resource Planning (ERP) systems, prove to hold valuable repositories of event data, encompassing essential business process footprints due to the process-centric nature of these systems (Yu Chung Wang et al., 2022). Leveraging such data enables the analysis of business processes on a larger scale, facilitating a comprehensive understanding of organizational workflows (Van der Aalst, 2008).

Process mining, as an emerging topic within the broader topic of business intelligence (BI) (H. Chen et al., 2012), is a data-driven approach that aims to extract valuable insights and knowledge from event logs (Van der Aalst, 2016). It involves the systematic analysis of event data to discover, monitor, and improve business processes. At its core, process mining leverages the digital footprints left by activities and transactions recorded in event logs within enterprise systems, such as Customer Relationship Management (CRM), ERP, or Supply Chain Management (SCM) systems (Van Der Aalst et al., 2007). These event logs capture the sequence of events, timestamps, and relevant contextual data that provide a detailed representation of actual process executions (Van der Aalst et al., 2011).

In summary, process mining enables organizations to gain insights into their processes by analyzing event logs generated by enterprise systems. By providing detailed and data-driven information about how business processes are performing, process mining contributes to the domain of organizational performance. The significant roles of process mining in this context are multifaceted. First, process mining enables organizations to gain an objective and accurate view of their processes, bypassing biases that may be present in manual process documentation

(Van der Aalst & Dustdar, 2012). Analyzing real data with process mining allows for building a complete and unbiased picture of process performance (Eggers & Hein, 2020).

Second, process mining aids in conformance checking, where the observed process behavior is compared against the intended process model (Carmona et al., 2022). This verification ensures that processes are adhering to predefined guidelines and compliance requirements (Becker & Buchkremer, 2019). Organizations make informed corrective decisions when they are aware of process deviations, thus effectively ensuring process integrity (De Medeiros et al., 2007).

Third, process mining contributes to process enhancement by offering insights into variations in process execution (Van der Aalst, 2012b). Organizations can identify best practices and successful process paths, paving the way for process optimization and standardization across the organization (Van der Aalst & Dustdar, 2012). Specifically, process mining uses the information stored within event logs. It generates insights on performance analysis, providing quantitative metrics and Key Performance Indicators (KPIs) to evaluate process efficiency, compliance, cycle times, and resource utilization (Badakhshan et al., 2022).

Fourth, process mining primarily centers on process discovery, a crucial set of methods to extract valuable insights from event logs and generate visual process models (De Leoni et al., 2016). A key feature of the event log is its ability to capture the control flow of process steps. This control flow represents the chronological order in which activities are performed within a business process for each case (Van der Aalst, 2016). Among many studies, Mans et al. exhibit how process mining can reveal the actual execution paths, variations, and deviations that occur in real-world processes by analyzing the control flow in event logs (2009). This unique aspect of the event log provides valuable insights into process performance and organizational performance (Badakhshan et al., 2022). Therefore, research can leverage the information embedded in the event logs to study complex topics such as organizational routines (Mahringer & Pentland, 2021; Wurm et al., 2021).

In short, studying process mining event logs enables organizations to gain business intelligence by visualizing and understanding their processes, identifying inefficiencies, and evaluating performance (Badakhshan et al., 2022; H. Chen et al., 2012; Reinkemeyer, 2020). Studying process mining event logs in combination with contextual data (e.g., production and error reporting data) provides more meaningful insights (Van der Aalst, 2012a). This contextual information adds relevancy to the analysis by highlighting the best practices, successful process paths, and potential areas for performance enhancement (Bose & Van der Aalst, 2009b).

This dissertation explores the intrinsic value of analyzing event logs to obtain business intelligence and enhance organizational performance. Notably, it acknowledges the previous contributions made in event log analysis, particularly within the area of process discovery, which studies event logs to understand business processes more deeply. Building upon these advancements, this research aspires to further delve into the untapped potentials latent in event logs, striving to unveil novel approaches that extract more comprehensive and profound insights from the data they encapsulate.

Moreover, this study augments its scope by leveraging the information extracted from event logs to contribute to the discipline of business intelligence and organizational behavior, with a particular focus on organizational routine, routine performance, and error management topics. Information systems scholars have already addressed this research gap, wherein, for example, Pentland, Recker, et al. (2020) underscored the potential inherent in analyzing event data to study organizational routines and develop theoretical underpinnings for information systems. This dissertation aims to extend and enrich the body of knowledge in the business intelligence and organizational performance domains by using event log analysis as a powerful tool to explore the dynamics of organizational behavior and performance.

In light of this introduction, this dissertation proposes the following overarching research question:

ORQ: *"How does studying event logs facilitate the acquisition of business intelligence and contribute to enhancing organizational performance?"*

To address the overarching research question, the present study embarks on an exploratory journey, comprising multiple steps (see figure 1) that collectively contribute to advancing the understanding and significance of event logs in the domain of process mining and organizational behavior. Accordingly, four papers were developed to address the aforementioned overarching research question.

Figure 1: A representation of the main topics discussed in this dissertation

The first paper, highlights the unique value inherent in event logs, particularly focusing on the control flow data structure encoded within the event log traces. By conducting an extensive review encompassing 101 papers on the topic of trace clustering, this paper introduces a generic framework for trace clustering, which encompasses critical aspects such as event log data, processing requirements, and the availability of domain knowledge. Through this comprehensive framework, the study sheds light on the substantial potential embedded within event log analysis.

In the second paper, a detailed investigation is conducted into the identification of unwanted patterns in process execution, with a particular emphasis on incorporating basic contextual knowledge about process steps. This study characterizes inefficient patterns within the process using the provided contextual information. Ultimately, this study proposes a process inefficiency index. The case study results in this paper underscore the rich information that each event log encapsulates in terms of process patterns, thereby reinforcing the significance of event log analysis for process optimization.

Moving forward, the third paper emphasizes the crucial role of capturing and analyzing patterns from event logs and seeks to offer a unified library of process measures previously introduced in process mining research (Zandkarimi, Decker, et al., 2021). This paper provides an invaluable analytical tool that facilitates and fosters further research and exploration of event logs, opening new avenues for innovative investigations.

Finally, the fourth paper explores the control flow of a production process event log, where the order of process steps for long-term orders remains constant. This unique setting presents an opportunity to design a natural experiment that examines performance changes, specifically related to performance efficiency and performance effectiveness, in response to errors. In this study, a crucial step was to separate the control and test groups for the experiment. However, achieving this separation necessitated the extraction of process execution paths from the event log. This process allowed for the identification and isolation of distinct groups, enabling a more rigorous analysis of the experimental variables. Similarly, when measuring the independent variables, namely throughput time and reporting accuracy, the event log data proved to be indispensable. The event log provided the necessary information to quantify these variables accurately and objectively, ensuring the reliability and validity of the findings. Thus, the utilization of event log data played a pivotal role in enabling the successful execution and interpretation of the experiment, ultimately enhancing the robustness of the research outcomes. The paper aims to uncover and elucidate the intricate dynamics that lead to performance changes in response to errors.

Through this multi-faceted approach, the study seeks to advance the frontiers of process mining by using the potential value encoded within event logs and exploring various aspects that contribute to organizational performance enhancement. Considering the distinct contributions of each paper, this research strives to enrich the academic discourse and contribute meaningfully to the field of process mining and its implications for organizational studies, specifically, organizational routines and organizational performance.

Table 1: Summary of each paper's contribution

Paper	Research gap	Contribution	Results
Paper 1	Due to the diverse range of algorithmic capacities, similarity functions, data characteristics, and computational complexities involved, the process of selecting the most appropriate trace clustering method becomes a non-trivial task.	Consolidate state-of-the-art trace clustering techniques and concepts. Highlight the unique structure of event log traces.	A generic framework for trace clustering.
Paper 2	Existing indicators represent a (too) generic inefficiency index, such as the presence of inefficient activities in a case, they fall short in leveraging the information encoded within the event logs, hence offering an incomplete understanding of process inefficiencies.	Identify inefficient patterns in event logs and introduce an inefficiency index based on basic domain knowledge about process execution rules.	An artifact to quantify various aspects of inefficiency within a process trace.
Paper 3	Previous studies do not refer to fully distinct and exclusive measures, leading to the repeated implementation of similar measures on different platforms.	Review the previous studies implementing event log measures.	A coherent library of 73 event log measures.
Paper 4	Lack of research on dynamics of production routines, which have unique characteristics, including shorter turnaround times, strict regulatory standards, fixed action sequences. The role of errors as interruptive events in shaping these routines remains understudied due to challenges in obtaining objective and quantifiable data. Empirical studies focusing on errors and their impact on organizational routines are encouraged to develop novel error management strategies based on changes in employee performance.	Investigate the effect of error communication on routine performance.	Routine efficiency and effectiveness are positively affected by errors but only for orders with fewer prior errors.

2 Research Papers

2.1 Introduction to Paper One

Paper one (Zandkarimi et al., 2020) presents a comprehensive study on the topic of trace clustering in process mining, which aims to simplify complex process models obtained from event log data (Bose & Van der Aalst, 2009b). Process discovery, a critical function in process mining, generates process models from event logs but often results in intricate and difficult-to-understand "spaghetti models" (Medeiros et al., 2007). Trace clustering is a powerful technique to address this issue, as it groups event log traces into cohesive sub-logs, allowing for the creation of less complex and more comprehensible process models. Over the last 15 years, numerous approaches to trace clustering have been proposed, each with varying algorithmic capacities, data characteristics, computational complexity, and integration of contextual information.

The paper's main contributions are a systematic literature review on trace clustering and the proposal of a generic framework for structuring and defining relevant components of trace clustering methods. The framework serves two purposes. First, it provides an overview of existing trace clustering techniques and methods, offering a methodological model. Second, it acts as a basis for advancing the field of trace clustering by identifying similarities with related data mining areas. This study suggests that the choice of the best trace clustering approach depends on the specific context, such as available data, data quality, and computational resources. Overall, this paper contributes to the advancement of trace clustering research and aids in the development of improved process discovery methods.

This paper has been published at the 2$^{\text{nd}}$ International Conference on Process Mining (ICPM). My role as the initiator and lead author included collecting and organizing the systematic literature review, developing interim findings, and shaping the main contribution of the paper, i.e., the generic framework. Collaborating closely with co-authors, I made significant contributions to crafting the manuscript. The culmination of our collective effort resulted in the paper's acceptance at the ICPM conference, where I presented our work to the community.

2.2 Introduction to Paper Two

Paper two (Zandkarimi, Rennemeier, et al., 2021) focuses on analyzing event logs to measure process inefficiency. Process efficiency is a significant component of organizational efficiency. Inefficient process design and implementation lead to improper resource allocation, hence missing organizational goals. The first trivial step in controlling process inefficiency is deploying measurement tools, i.e., process performance measurement. An event log includes data about each process step. Hence, potential inefficient patterns can be identified, quantified, and monitored based on the event log data.

Most industrial process mining tools offer an inefficiency index to measure process inefficiency based on event log data. The paper reviews the previous studies on inefficiency and highlights the inefficiency nuances that are overlooked by the current industry tools. The article proposes an artifact that implements the previously missing aspects of process inefficiencies into a new process performance measurement index. The introduced index captures more inefficiency dimensions than previous approaches, e.g., measuring the distance between occurrences of non-value-added activities. The final results indicate that the proposed approach captures aspects of process inefficiency previously overlooked by existing tools. Ultimately, the findings trigger interesting questions about process inefficiency, e.g., whether process inefficiency is an institutionalized phenomenon that generally does not tend to change rapidly.

Serving as the primary initiator and lead author, I closely collaborated with my co-authors, offering substantial support in developing the artifact and conducting an extensive literature study on nuances of process inefficiency. Our collective efforts resulted in the submission and acceptance of the manuscript at the 19th International Conference on Business Process Management (BPM), where I presented our work to the community.

2.3 Introduction to Paper Three

In process mining projects, event log measures play a vital role in characterizing event logs and providing valuable insights for various process mining applications. These measures, numeric representations of raw data, are calculated at the trace level and aggregated to represent event log characteristics. However, the multitude of process mining applications often leads to the re-implementation of event log measures across different platforms. To address this issue, we developed Fig4PM, an open-source application, to establish a standard, comprehensive, and reusable library for calculating event log measures (Zandkarimi, Decker, et al., 2021). The current version of Fig4PM offers 73 distinct control-flow measures, derived from the literature or existing measures. The ultimate goal is to create a public Python library that facilitates feature generation in process mining applications, reducing redundancy and enhancing research efficiency.

The relevance of event log measures in process mining is evident through their application in various studies and applications, including data preprocessing, data quality assessment, predictive process mining, deep learning techniques, business process simulation, process complexity analysis, and trace clustering. To avoid duplication of effort and promote collaboration, the Fig4PM library serves as a repository where researchers and practitioners can access previously implemented event log measures. It acts as a starting point for further development, enabling users to contribute new measures, enhance existing functions, and improve overall performance. By fostering a standardized approach to event log measures, Fig4PM aims to streamline the process mining workflow and advance research in the field, ultimately facilitating better data analysis and insights for process improvement and decision-making.

This paper has been published at the 3rd International Conference on Process Mining (ICPM). My role as the initiator and lead author included supporting the development of the artifact, collecting the measures, and publishing the final work as a public library. Collaborating closely with co-authors, I made significant contributions to crafting the manuscript. Eventually, the paper accepted at the ICPM conference, where I presented our work to the community.

2.4 Introduction to Paper Four

Paper four investigates the impact of errors on the performance of frontline employees in the context of production routines. The study presents a unique contribution by employing a novel dataset derived from production routines within a large Dutch public holding company. Through a comprehensive analysis of the 75,000 reported errors associated with 3 million production batches over six years, the study aims to understand how errors influence post-error improvement in accuracy (PIA) and post-error slowing (PES) in organizational routines. This study finds evidence that production teams tend to work more accurately and faster after errors are reported, although this effect diminishes with increasing error frequency.

Theoretical implications of this study extend to the field of organizational routines and error management. The study contributes to routine dynamics by showcasing how errors can shape and transform employees' actions within an inflexible environment. This unique perspective contributes to the discourse on the adaptive nature of organizational routines and their responsiveness to external influences. Moreover, the research underscores the significance of examining the temporal aspects of error effects, filling a gap in error management literature. The findings also emphasize the need for further research on error-induced performance changes in varying organizational contexts and the impact of other factors like employee turnover.

Practical implications stemming from this research underscore the critical role of errors as potential sources of innovation and improved performance within organizational routines.

Managers are advised to recognize errors not merely as disruptions, but as opportunities to drive positive changes in employees' behaviors. It is imperative for management to strike a balance between error prevention and effective error management strategies. Although occasional errors can drive short-term performance enhancements, addressing recurring mistakes promptly becomes crucial to sustaining productivity gains and upholding customer satisfaction. By acknowledging the nuanced interplay between errors and performance, managers can tailor strategies that use errors' positive potential while minimizing their adverse impacts in the long term.

Overall, the paper emphasizes the significance of errors as triggers for adaptive changes in organizational routines and highlights the intricate relationship between error management and sustained performance improvement in dynamic work environments.

As the lead author, I was responsible for several activities. Firstly, I managed communications with the industry partner, ensuring secure and effective data access. This involved collecting metadata to understand their production processes as well as reporting our research goals to the management team. Secondly, I developed a large-scale data preprocessing and cleansing pipeline to address various data quality issues we encountered. Thirdly, I conducted the extraction of the entire process event log, a massive data transformation process necessary for building all the statistical analyses later presented in the paper. I played a key role in formulating the theories and crafting the overall manuscript, which was completed with the support of my co-authors in conducting the final statistical analysis and writing up the paper.

Together with my co-authors, we have chosen to submit this paper to the Information Systems Research (ISR) journal, as ISR is known for its emphasis on rigorous empirical research in the field of information systems. By employing event log data to conduct a controlled experiment on performance changes in response to errors, this study exemplifies the kind of methodological approach and empirical depth valued by ISR. Its contribution to understanding the intricate dynamics of organizational routines aligns with the journal's emphasis on advancing the understanding of information systems' impact on organizations performance.

Table 2: Overview of the studies presented in this dissertation.

	Paper 1	**Paper 2**	**Paper 3**	**Paper 4**
Title	A Generic Framework for Trace Clustering in Process Mining	Are We Doing Things Right? An Approach to Measure Process Inefficiencies in the Control Flow	Fig4PM: A Library for Calculating Event Log Measures	How Errors Shape Production Routines: An Empirical Examination of Digital Traces in Manufacturing
Analysis focus	Technical aspect of process mining control-flow	Technical and business aspect of process mining control-flow	Technical aspect of process mining control-flow	routine dynamics, error management, habits theory
Data Source	Literature review	BPI 2019, BPI 2020	Literature review	6 years, 3M production steps, 75K complaints
Publication status	Published: International Conference on Process Mining (2020)	Published: International Conference on Process Mining (2021)	Published: Business Process Management Forum (2021)	Ready to submit: Information System Research
Contributors	Zandkarimi, Rehse, Soudmand, Hoehle	Zandkarimi, Rennemeier, Rehse	Zandkarimi, Decker, Rehse	Zandkarimi, Pethig, Hoehle, Sabherwal

3 Paper 1: A Generic Framework for Trace Clustering in Process Mining

3.1 Abstract

The goal of process discovery is to visualize event log data as a process model. In reality, however, these models are often highly complex. Process trace clustering is a well-studied and powerful technique to address this. It groups an event log into more cohesive sub logs, such that the discovered process models become less complex and easier to understand. Over the past 15 years, researchers proposed various approaches for trace clustering in process discovery. The developed approaches vary greatly with regard to algorithmic capacities, data characteristics, computational complexity, and integration of additional information. In this paper, we provide a state-of-the-art analysis of trace clustering by a) performing a systematic literature review, and b) proposing a generic framework for trace clustering. Eventually, our goal is to provide an overview of current trace clustering research and a basis for developing new methods and approaches to trace clustering.

3.2 Introduction

Process discovery is one of the three main functions in process mining (PM). Its main goal is to visualize a real-life business process, as recorded in an event log, in a human-readable way (Van der Aalst, 2016). Therefore, discovery approaches generate process models, which are supposed to provide their users with a graphical representation of the examined process. In reality, however, these approaches often produce so-called spaghetti models, i.e., highly complex models that are difficult to read and understand (Medeiros et al., 2007). This is particularly relevant because process discovery is typically the first step in a process mining project and the resulting models are used for many purposes, such as conformance checking or process simulation.

Hence, improved process discovery results may lead to an overall improvement of process-based decisions made by humans and/or machines.

One way to achieve those improved results is to preprocess the event log before applying process discovery in order to decrease the process model complexity. For this purpose, process trace clustering is a well-known and effective technique (Bose & Van der Aalst, 2009b). It groups the traces in the log, such that the traces in each group (called cluster) are more similar to each other than to those outside the group, keeping the clusters as distinct as possible (Reijers et al., 2011). Applying process discovery on those more cohesive sub logs results in less complex and better readable models, which do not represent the entire event log at once. For example, the log of a hospital emergency room will contain multiple process variants, depending on the urgency, diagnosis, and treatment of the individual patients (Lu et al., 2019). In order to analyze the process effectively, trace clustering can divide the log along with these attributes, such that the process flow for each group of patients can be examined individually.

Over the past 15 years, researchers proposed various approaches to apply trace clustering in the context of process discovery. The developed approaches vary greatly with regard to algorithmic capacities (e.g., density-based or hierarchical clustering), similarity functions (e.g., activity-based similarity), data characteristics (e.g., event log attributes), computational complexity, and integration of external knowledge (e.g., domain expertise). This makes trace clustering a context-specific task, i.e., the choice of the best approach depends on the availability of data, quality of the data, or available processing power, among others.

In this contribution, we aim to structure the existing body of knowledge in trace clustering. Therefore, we perform a systematic literature review to identify and summarize the current state-of-the-art. Based on that, we propose a generic framework for defining and structuring relevant building blocks of trace clustering methods. The goal of the framework is twofold. It summarizes the current trace clustering techniques and methods in the form of a methodological model and it provides a framework for further developing the field of trace clustering by identifying similar techniques in the related data mining areas. Therefore, we first summarize the background of clustering in process mining in Section 3.3. Section 3.4 describes our research method. The results of our literature review and the proposed framework are covered in Section 3.5. Section 3.6 discusses the results and concludes the paper.

3.3 Background

Cluster analysis is a well-established data mining technique aimed at finding groups of similar items in a dataset (Landau et al., 2011). It is typically categorized as an unsupervised machine learning technique (Berkhin, 2006). Clustering techniques are widely used for analyzing sequential data, e.g., sequence analysis techniques in bioinformatics, sequential pattern mining for user buying patterns, and sequence labeling for part of speech tagging. In the PM context, it can be applied to many different data objects, including events, activities, and process models. Arguably, however, the most relevant application is the application of clustering techniques on process event logs, which we focus on in this contribution. Trace clustering is often used as a preprocessing technique to improve process discovery results. For example, it can help to find statistical outliers and reduce noise in an event log (Fani Sani et al., 2018; Weijters & Aalst, 2001), but it may also support predictive monitoring of business processes by finding predicates that a running instance will most likely fulfill (Di Francescomarino et al., 2016).

We define event logs as collections of traces that represent the behavior of a business process and assume that each trace has one mandatory attribute (the case identifier) and consists of a sequence of events, denoting the execution of activities. Each event also has mandatory attribute (the activity identifier). We call case identifiers, activity identifiers, and the ordering of events in a trace the control-flow perspective. If traces or events have additional attributes, such as timestamps, cost, and resources, we address those as the context perspective.

Trace clustering as a preprocessing technique for process discovery has been researched for more than a decade (Greco et al., 2004) and has experienced multiple developments regarding the techniques, similarity measures, computational complexity, and maturity. The morphological box by Thaler et al. (2015). gives a good overview of the approaches published before 2015 (Thaler et al., 2015). Early approaches to trace clustering treated traces as bags of activities, losing information on context or execution order (Greco et al., 2006; Medeiros et al., 2007). This was addressed by using different similarity measures such as the generic edit distance (Bose & Van der Aalst, 2009b), sequences (Veiga & Ferreira, 2010), or temporal proximity (Luengo & Sepúlveda, 2012). All of those approaches are two-staged, such that the discovery results are not considered during clustering. This problem is addressed by considering the properties of the discovered model during clustering (De Weerdt et al., 2013), mining more accurate process variants or sub-processes (García-Bañuelos et al., 2014), or finding a more appropriate distance measure (Evermann et al., 2016).

Given the plethora of technically mature approaches, current research on trace clustering focuses on making the results more accessible for process analysts. De Koninck, De Weerdt, et al. (2017) describe an approach for explaining the assignment of traces to clusters and a new technique for trace clustering that incorporates expert knowledge (De Koninck, Nelissen, et al., 2017). Seeliger et al. (2019) present the ProcessExplorer tool, which is set out to support the typical workflow of a process analyst to interactively explore a dataset.

Although a large body of research on trace clustering already exists, we are currently unaware of any generic framework, which would help to systematically assess the body of knowledge and the gaps in the field.

Table 3: Systematic literature review

Round	Search.Criteria	Time	Results
1	”clustering” + ”process mining”	2000 - 2020	5860
2	(”trace clustering”) + (”process mining”) [”trace clustering” clustering OR process OR processes OR traces ”process mining”]	2000 - 2020	691
3	English papers only	2000 - 2020	630
4	Relevance according to titles; removing case studies, literature reviews, books, and short papers	2006 - 2020	126
5	Relevance according to abstract and content, removing duplicates	2006 - 2020	70
6	Forward and backward search	2006 - 2020	103

3.4 Research Method

In order to design such a generic framework for trace clustering in process discovery, we first conducted a systematic literature review (Webster & Watson, 2002), summarized in Table 3.

We conducted our search using Google Scholar, which contains, among other databases, IEEE Xplore, SpringerLink, and the ACM Digital Library. Our final search string was designed to capture studies mentioning trace clustering and process mining. We started without any further filters in the first round, and then restrained the search terms and filters after each iteration. Keyword search inside the whole text for peer-reviewed publications in English returned 630 studies. We filtered out duplicates, books, short papers, datasets, and irrelevant studies (based on title, abstract, and keywords). The resulting set contained 126 studies, which we filtered based on their content. 70 papers were kept as relevant. We then conducted a forward and backward search on those papers to cover studies missed by term search (Webster & Watson, 2002). Finally, a set of 103 papers was used to conduct this literature review. Table 4 shows the distribution of papers over the years.

We used the 103 papers found in the literature review to design a generic framework for trace clustering in process mining (see: Appendix A). First, we identified common concepts and

Table 4: Distribution of the collected studies per year.

Year	**Results**
2004-2011	16
2012	4
2013	7
2014	7
2015	11
2016	11
2017	18
2018	11
2019	11
2020	7

building blocks that constitute different trace clustering approaches. Then, we focused on the ordering and dependencies of these building blocks. This included explicating the data flow between them and differentiating between necessary and optional steps. At this point, we had a framework that illustrated the generic procedure of trace clustering as found in the current literature. We then extended this framework with additional building blocks that were not found in the process mining domain, but were established and promising techniques that are used for cluster analysis in data mining. The purpose of this step was to get a more complete overview of trace clustering and to show opportunities for future research. Finally, we went back to the 103 papers and identified the concepts and techniques that they used to realize each generic building block. This framework is a methodological model, as it is derived from studying trace clustering methods and reflects the possible method development paths. We also argue that this model is mostly descriptive (relying mostly on the given literature) and partly normative (due to introducing new building blocks).

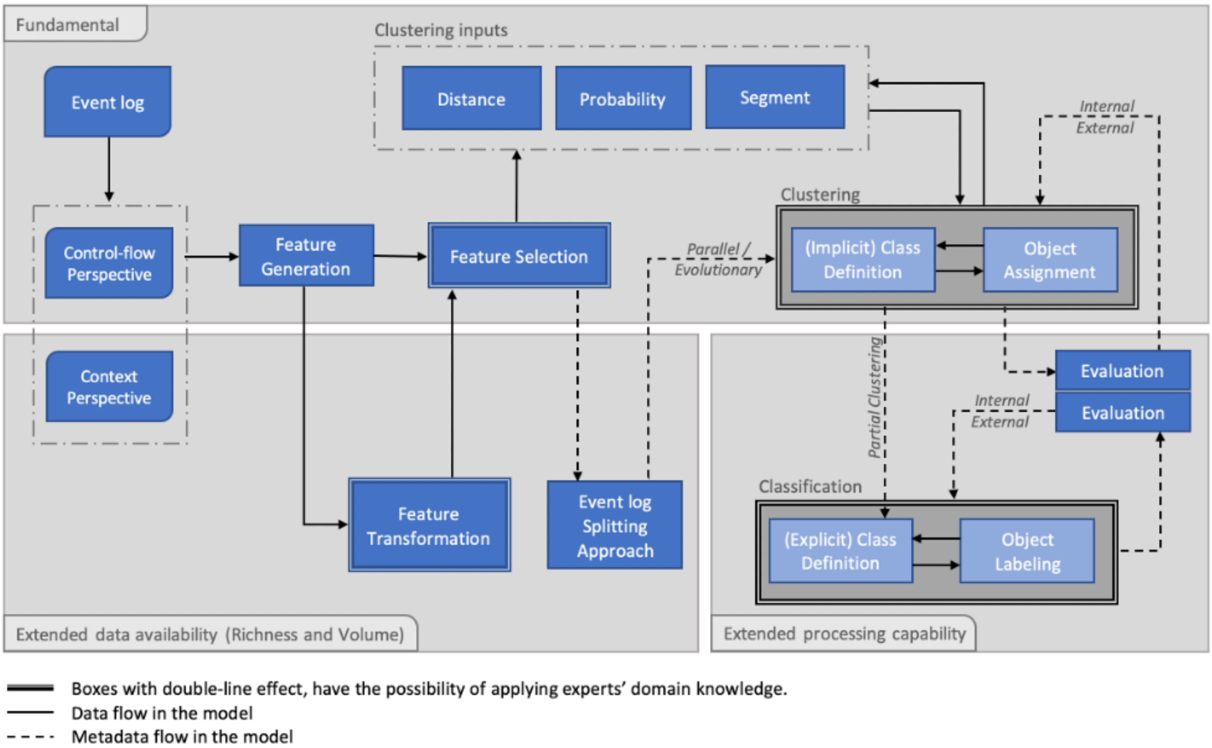


Figure 2: Generic framework for Trace Clustering in Process Mining

3.5 A Framework for Trace Clustering

3.5.1 Overview

Figure 2 shows our proposed generic framework for trace clustering in process discovery. It consists of blue boxes, which describe the generic building blocks of trace clustering approaches. One of the first things we noticed in analyzing the trace clustering papers was that many of these building blocks depend on the availability of either certain attributes in the event log to measure trace similarity or certain processing capabilities to execute computationally complex clustering algorithms. Therefore, we divided the building blocks into three major groups, indicated by the grey areas in the framework. Building blocks in the "fundamental" area can be applied in any trace clustering context, where a basic event log and standard processing capabilities are present. Building blocks in the area "extended data availability" can only be applied if the event log fulfills certain properties regarding data richness and volume. This may refer to the presence of a certain attribute, the number of attributes, or the amount of traces in the log (Leyer, 2011).

Finally, building blocks in the "extended processing capabilities" area can only be applied if large computational resources are available, because standard capabilities are not able to apply computationally complex algorithms to large event logs.

The building blocks of the framework are connected by arrows, which represent their ordering in the trace clustering process and the flow of data between them. The solid-line arrows show the flow of event log data, while the dotted arrows are signs of exchanging metadata, e.g., signals or parameters.

We consider domain knowledge as an optional input to trace clustering, which can be utilized to different building blocks. These blocks are marked with a double-line effect meaning that during our literature analysis, we found the potential of applying domain knowledge in that particular step.

Each trace clustering project starts with an event log, which is represented by the building block in the top left corner. As explained in the background section, we consider two general perspectives on the event log, depending on the available data. The control-flow perspective only refers to case identifiers and sequences of activity identifiers (i.e., events). All possible paths in the fundamental area are applicable to such a basic event log. If a given event log contains additional attributes, e.g., resources, timestamps, or cost, these are represented by the context perspective. In this case, the blocks in the "extended data availability" area become applicable to this event log.

All available attributes in an event log can be mapped to either of the two following possible granularity levels: activity-level (e.g. cost of activity, department, user) and trace-level (e.g. order type, order duration, customer satisfaction). In terms of data types, all attributes in the event log are either numeric or categoric data. Transforming these two data types into each other occurs in the succeeding steps.

Table 5: Two examples of treating categoric and numeric data types

Inout	Matrix	Vector	Scalar
$< abac >$	$\begin{array}{c} a \\ b \\ c \end{array}\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}$	$\sqrt{6} \approx 2.45$
$< 15, 12, 8, 5 >$	$\begin{array}{c} a \\ b \\ c \end{array}\begin{bmatrix} 15 & 0 & 8 & 0 \\ 0 & 12 & 0 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}$	$\begin{bmatrix} 23 \\ 12 \\ 5 \end{bmatrix}$	$\sqrt{698} \approx 25.42$

In the following sections, we discuss the framework's building blocks in more detail. A summary of the concepts and techniques used to realize them is presented in Table 7.

3.5.2 Feature generation

The first step towards trace clustering is to generate features from the provided event log (Song et al., 2009). A feature can be any property or attribute contained in the event log. As the overall goal is to assess the similarity between individual traces, features will be generated at trace or event level. According to our observations in the literature, two data structures are typically used for feature generation. We categorize them as linear structures and non-linear structures. Linear features typically refer to simple data structures, such as scalars. Non-linear structures, e.g. graph and tree structures are used when the non-linear behavior of traces cannot be fully captured by those linear data structures. Depending on the required features, numeric or categoric data from the event log may get transformed from one type to another. In the following, we provide examples for such transformations.

Assume a sample trace $< abac >$ with the following durations $< 15, 12, 8, 5 >$ associated with the respective events. For this trace, we can generate a new feature with a different data type by mapping any duration value lower than 10 to low (L) and values higher than 10 to high (H), resulting in the (linear) feature $< H, H, L, L >$. This is a simple example of converting numeric data type to categoric.

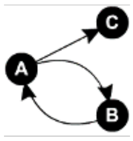
Table 5 provides an example of three possible representations of a trace $< abac >$ with durations $< 15, 12, 8, 5 >$. This trace can be represented by a two-dimensional array (matrix), a one-dimensional array (vector), or a single value (scalar). Table 5 shows how those numeric features are extracted from a categoric (first row) and a numeric (second row) input. In both examples, the rows in the matrix represents the activities and the columns represent the positions in the trace, meaning that activity a appears both in the 1st and 3rd position and that the duration of the 2nd activity is 12. Depending on the required level of information abstraction, different transition functions (e.g., count, sum, mean) can be applied to transform these matrices into new data structures. In our example, the rows are summed up to transform the matrix into a vector. This reduces the feature size and therefore its sparsity, but also loses information on the trace ordering. The same happens when the vector is transformed into a scalar by applying a magnitude (sum of squares) function (Luengo & Sepúlveda, 2012).

Moving from matrices to scalars results in a higher abstraction level. A similar observation would be moving from activity-level to trace-level features. The provided example in Table 5 shows individual activities of a trace, but one can consider the same features for any given subsets of a trace.

A subset of trace t is also known as an n-gram, where $0 < n \leq length(t)$. For example, $< ac(0) >$ is a 2-grams of $< abac >$ that represents the occurrence of a followed by c with a distance of zero. Two principal approaches can be considered when processing an event log based on n-grams: manual (brute force) and algorithmic approach (explorative algorithms) (Greco et al., 2006). The manual approach tries all possible n-grams and extracts features from them, whereas in the algorithmic approach, only significant n-grams, such as the most frequent ones, are used for feature generation (Bose & Van der Aalst, 2010b).

In our example, the underlying data structure is linear, i.e. we can interpret the trace as a sequence of activities that can be cut into subsets of different sizes. However, one could also

Table 6: Conceptualizing a trace as a graph

Graph	Matrix	Incoming	Outgoing
	$\begin{array}{cc} & \begin{array}{ccc} a & b & c \end{array} \\ \begin{array}{c} a \\ b \\ c \end{array} & \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array}$	$\begin{array}{cc} \begin{array}{c} a \\ b \\ c \end{array} & \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \end{array}$	$\begin{array}{cc} \begin{array}{c} a \\ b \\ c \end{array} & \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix} \end{array}$

conceptualize a trace as a graph structure, highlighting the existence of a link between any pairs of activities (Diamantini et al., 2016; Ha et al., 2016). In this case, we can generate new features by, e.g., discovering isomorphic subgraphs. As an example of the same trace ($< abac >$), the degree vector would be $< 2, 1, 1 >$ for the respected undirected graph. For the directed graph, we can consider $< 3, 2, 1 >$ as the degree vector, $< 2, 1, 0 >$ as the outgoing vector, and $< 1, 1, 1 >$ as the incoming vector (see Table 6).

3.5.3 Feature transformation

The generated features of the previous step provide the input for the next one. Feature transformation includes generating secondary features and assuring feature quality and interpretability by applying techniques for normalization, collinearity, and dimensionality. Normalization ensures the comparability of features. Collinearity checks for redundancy, i.e., eliminating identical or similar features. Dimensionality ensures a certain degree of richness, so multiple versions of one attribute exist in the final pool of features, so if feature selection removes certain attributes, not all information is lost (Song et al., 2013).

Secondary (or transformed) features are generated by applying linear techniques, e.g. wavelets (Taymouri, Rosa, et al., 2020), principal component analysis (PCA), factor analysis (Bartl et al., 2011), or non-linear techniques (machine learning (ML)-based and non-ML-based).

3.5.4 Feature selection

In the last step of the pre-clustering phase, the generated and transformed features are reduced by feature selection techniques (Greco et al., 2004). Those techniques are needed to avoid problems with overfitting, precision issues, and extra processing costs, which can appear when using too many features. However, the feature selection process itself needs processing resources, too. It happens either in an all-at-once (single-view) or incremental (multiple-view) fashion (Appice & Malerba, 2015). The former usually yields relatively fewer features, whereas the latter allows for selecting more features in the final set.

Generally, feature selection techniques can be split into two categories, depending on whether they are based on machine learning (ML). Non-ML-based techniques include for example entropy-based filtering (De Weerdt et al., 2013), filtering collinear features (S. Lee et al., 2013), and frequency-based selection. They also allow to integrate experts' domain knowledge into the feature selection process.

The ML-based techniques can be divided into two main approaches, embedded and wrapping. The embedded approach is a built-in feature selection mechanism that occurs during some trace classification algorithms (Cuzzocrea et al., 2018). It refers to the selection of significant features during trace classification, which we call partial classification. After processing a subset of traces, a set of clusters is defined, to which the remaining clusters are then assigned (Di Francescomarino et al., 2016). Features are justified based on the model performance, i.e., significant features have better accuracy in assigning clusters. The wrapping approach, on the other hand, tests different combinations of features in order to find the most significant ones. Features are justified based on the evaluation results of ML techniques (Genga et al., 2020).

3.5.5 Event log splitting approach

Dataset splitting is a common approach in data mining field mostly to handle complex and large data volumes (Berkhin, 2006; Zhao et al., 2019), which can become relevant for trace clustering as well. Dealing with large event logs can be computationally expensive and time-consuming. Processing the whole event log, however, does not guarantee optimal results. Hence, depending on the dataset, technique, and available computing resources, it might make sense to preprocess the event log by splitting it prior to clustering. Principal approaches in dealing with large event logs can be labeled as parallel and evolutionary. In both approaches, the event log is split into multiple sub logs.

The parallel approach exploits clustering and classification techniques to process all sub logs simultaneously. The applied settings and algorithms can vary for each sub log, to ensure the distinctness of the results. Processing each sub log yields a set of sub-clusters, which are merged to obtain the final clusters. In the evolutionary approach, randomly sampled sub logs are processed sequentially, until either all sub logs are processed or the updated clusters do not change significantly.

None of the approaches that were found in the literature review had implemented event log splitting prior to clustering. However, we included it in the framework, because it's an established technique in data mining, which we propose and recommend for future research, especially when dealing with very large event logs.

3.5.6 Clustering inputs

Clustering algorithms aim to group similar traces such that traces inside the same group are similar and traces from different clusters are different. The way in which trace (feature) similarities and differences are measured is therefore central to the resulting clusters. Distance measures are a popular technique for measuring trace similarity. Using them, we can compute a so-called similarity matrix, which contains pairwise similarity values for all traces in a log and serves as input for some clustering algorithms. Other potential inputs for clustering algorithms are probabilistic models and segments. These three potential building blocks of trace clustering are shortly discussed in the following.

3.5.6.1 Distance There are multiple ways to calculate the distance between two traces (or their respective features), which vary according to distance type, granularity, and robustness function. The distance type can either be string-based, accepting all types of features, or arithmetic, accepting only numeric features. Calculating arithmetic distance requires mathematical operations which is not the case for string distance. Instead, string-based distance measures quantify the differences between two sequences of features of any type. One way to compute these differences is the edit distance, which refers to the number of operations (insert, delete or move) required to make two given sequences completely similar. In case of actual strings, this is also known as the Levenshtein distance (Levenshtein et al., 1966). Other string distance functions use different methods to calculate these differences. Behavioral distance is applied for features with a tree-like data structure. Morphing-based functions are used to compare traces. The base-model distance approach discovers an initial model, e.g. a graph, based on limited or all traces. This base-model calculates pairwise edge distances for respective nodes of the traces on the model and updates the results (Diamantini et al., 2016).

So far, we assumed that all approaches compare traces to model their differences. However, the same is possible by changing the unit of processing to batches of traces, i.e., segments (Ceravolo et al., 2017). Transition functions can build new “per segment” features. One last optional part of this step is implementing robustness functions. These functions help to moderate inappropriate distance values, happening due to the presence of erroneous data points (e.g. noise and outlier) in the event log (Delias et al., 2015).

Table 7: Underlying techniques and concepts of the proposed framework

Event log	**Feature Generation**	**Feature Transformation**	**Distance measure**
- Data availability: control-flow perspective (case identifier, activity identifier, order), context perspective (optional) - Data granularity: activity-level (e.g., duration, user, resources), trace-level (e.g., throughput time, quality, success) - Data type: categoric, numeric	- Data structure -- Linear: matrix, vector, scalar -- Non-linear: graph, tree - Data type: categorical, numerical - Transition function: magnitude, mean, max, frequency, etc. - Length: 1-gram, …, n-gram - N-gram building: manual, algorithmic - Granularity: activity-level, trace-level	*Linear* wavelet, PCA, factor analysis *Non-linear* - ML-based -- SOM, Deep belief Network - Non-ML-based -- kernel PCA, principal curves, Laplacian eigenmaps, diffusion maps	*Type* - String-based -- edit distance --- data structure: linear, graph --- distance function: Jaccard, hamming, levenshtein, etc. -- behavioral distance --- data structure: tree --- distance function: morphing -- base-model distance --- data structure: linear, graph --- distance function: geodesic - Arithmetic -- data structure: linear, graph -- distance function: Euclidean, cosine, google distance *Robustness function* - Moderation functions *Granularity* trace, segment
Clustering *Input type* - Distance-based hierarchical (bottom-up), density-based, partitioning - Probability-based - Segment-based grid-based clustering, hierarchical (top-down) *Assignment type:* hard, soft	**Evaluation** - Feedbacking: internal, external - Criteria: PM-related (improving clustering), non-PM-related (business use cases)	**Feature selection** *Mode* - Incremental, all at once *Type* - Non-ML-based expert-knowledge, entropy, frequency, collinearity - ML-based embedded, wrapping	

3.5.6.2 Probability Another option for trace clustering is to use probabilistic computation models instead of deterministic ones. In this approach, Markovian probabilistic models are built. The probability of observing a trace given the Markovian probabilistic model(s) determines to which cluster each trace belongs (Ferreira, 2009; Huang et al., 2015; Veiga & Ferreira, 2010).

3.5.6.3 Segments The third option in clustering inputs are segment-based computation techniques. In this approach, a conceptual grid is built, either gradually or all-at-once. In the all-at-once method, each segment together with its neighbors is tested to see whether merging them supports the goal of clustering or not (Kanj et al., 2018). The gradual model starts with the whole event log. In each iteration, the event log is split into segments until the clustering algorithm decides to stop. This decision is made based on the expressive power of clusters to satisfy the clustering goal. The final segments are basically the final clusters (H. Nguyen et al., 2019).

3.5.7 Clustering and classification

Clustering input blocks provide the necessary input data for clustering and classification algorithms. The clustering algorithm triggers new rounds of building the clustering input(s), iteratively. The provided input(s) are used by the clustering algorithm directly (e.g. k-means) or with further transformations (e.g. spectral clustering). We considered three main approaches for clustering, namely full clustering, full classification, and partial clustering. In the full clustering approach, detecting the initial clusters (implicit class definition) as well as assigning all traces to those clusters are all done mainly by the clustering algorithm. This approach also allows for integrating experts' knowledge when defining the initial clusters (Lu et al., 2019). In the full classification approach, expert knowledge (in the form of manual grouping or to-be process models) is used to define the initial classes (explicit classes) that are later used by classification techniques to assign the rest of the traces to the clusters (Boltenhagen et al., 2019). In the partial clustering approach, a combination of clustering and classification techniques is used. Clustering techniques, with the possible help of experts' knowledge, are applied to build the initial clusters based on a limited part of the event log. Eventually, classification part takes the final clusters as its initial classes and continues with processing the rest of the event log.

All three presented clustering inputs in the framework, i.e. distance models, probabilistic models, and segment-based models, provide input to feed classification and clustering algorithms (except segment block that cannot feed classification algorithms). Clustering building blocks and clustering techniques can be matched as follows. Segment-based matches with grid-based and top-down hierarchical clustering (Medeiros et al., 2007). Distance-based matches with hierarchical (bottom-up) (Seeliger et al., 2018), density-based (La Rosa et al., 2015), and partitioning (Delias et al., 2015). Probability-based clustering methods can be categorized based on their kernel methods (e.g. Gaussian, non-Gaussian) as well as their estimation methods (e.g. Bayesian, non-Bayesian).

In terms of cluster borders, clustering techniques divide into two categories, one allowing for overlapping (soft) and the other one not (hard). We observed implementing probability-based approaches for soft cluster splitting, however fuzzy approach is not implemented in any of observed trace clustering studies, so it remains as our suggestion for future research.

3.5.8 Evaluation

Usually classification and clustering algorithms have their own built-in objectives. However, we can also define extra objectives (evaluation criteria) depending on trace clustering's ultimate goal, e.g. enhancing process model discovery, demands for higher model simplicity, and fitness as evaluation objectives (Thaler et al., 2015). When a paper is not focused on making a methodical contribution to trace clustering or process mining, e.g., in case studies, usually standard built-in objectives suffice (we refer to this as non-PM-related evaluation criteria).

The purpose of defining extra objectives is assisting clustering and classification algorithms (PM-related evaluation criteria). This is achieved by generating feedback in two different ways, namely internally and externally. Internal feedback is generated per each iteration of running the algorithms (internal) (De Weerdt et al., 2013), while external is generated once, i.e. after finishing the algorithm (external) (Delias et al., 2015).

Internal feedback improves the precision of assigning traces in each iteration of running classification and clustering algorithms. External feedback optimizes the parameters of classification and clustering algorithms, e.g. number of clusters, nominated set of features, weights of each building block, etc.

It worth mentioning that clustering and classification algorithms can have multiple objectives (Delias et al., 2019). There are several techniques to handle this situation based on multi-objective decision making (MODM) approaches, e.g. weighted sum, disaggregation method, goal programming, and Pareto optimality.

3.5.9 Demonstrating a sample study in the framework

In order to demonstrate the validity of our framework, we briefly introduce an individual study (Delias et al., 2015) and map its components to our proposed framework. The authors exclusively work with the control-flow perspective of the event log. In the feature generation step, for each trace, they generate two trace-level numerical features in the form of vectors. Two transition functions generate these vectors based on appearance of each activity in the trace (1-gram subsets) and pairwise inversed sequence-distance of activities (2-gram subsets). In terms of advanced feature generation, we did not observe any relevant actions. The authors applied a cosine similarity distance function on both available features to generate two similarity matrices, namely activities-similarity and transitions-similarity (we refer to as clustering inputs). In this approach, neither event log splitting nor classification techniques were implemented. Spectral clustering algorithm applied two different weights (determined by the help of experts' knowledge) to each similarity matrix and then combined them into a single matrix using weighted sum function. We observed exploiting a robustness function (density-based weighting) to reduce the effect of outlier and noises in the data. In the absence of PM-related goals, no evaluation (internal, external) loops were presented in this study.

3.6 Conclusion

Trace clustering has been a topic of interest in process mining research for almost two decades. In this context, the goal of this paper was to structure the existing body of knowledge and give a comprehensive overview on the state-of-the-art in trace clustering. Therefore, we first performed a systematic literature review, in which we identified 103 relevant research works on trace clustering between 2004 and 2020. We then used these works to design a generic framework on trace clustering in an iterative, bottom-up way. This framework consists of 15 building blocks, which are grouped by applicability. The flow of data and metadata between those building blocks represents the generic process of trace clustering. In a second step, we analyzed how the 103 identified research works realized some of those building blocks, leading to a conceptual and technical overview of trace clustering capabilities, as shown in Table 7.

However, our research also suffers from multiple limitations. Despite following a methodical approach, we do not claim that our literature review or the framework are complete or exhaustive. Our choice of search terms or databases could have excluded relevant research works, which either used a different terminology or were not contained in the respective databases. Hence, there could be unidentified contributions to trace clustering, which could add additional aspects to our framework. The framework in its current form insinuates a generic process of trace clustering, which we have found to be true for many existing approaches, but which new approaches to trace clustering do not necessarily have to follow.

In addition, the framework's current building blocks result from our understanding and interpretation of the existing literature and could be conceptualized differently by other researchers. For example, one could argue that there are more than two perspectives to look at an event log or that trace classification is not part of trace clustering in a narrow sense. Also, our list of concepts and techniques that realize the respective building blocks can also not seen as complete, because it's highly likely that we missed or misclassified some approaches, for example due to different parametrization options. Our framework also falls short with regard to the generic data mining aspects and will require a more thorough argumentation on how they could address apparent gaps in trace clustering.

The overall goal of our generic framework is twofold. For practitioners, we want to give an overview over the current state-of-the-art in trace clustering and take a first step towards building the big picture of trace clustering techniques and implementation approaches. Categories like "extended data availability" and "extended processing availability" may help in this regard. However, the framework in its current form lacks practical utility, because it focuses more on the generic trace clustering process than on the capabilities of individual approaches. In addition, it has not been empirically validated by, e.g., other researchers or practitioners. In future work, this can be addressed by asking authors of our identified papers to position their work within our framework.

For researchers, we want to make it easier to position their own research with regard to the state-of-the-art in trace clustering. Although this is not the main focus of this paper, several of these gaps already became evident during our analysis. For example, there are no works on the "cluster-ability" of an event log, i.e., the attributes that an event log has to fulfill for trace clustering to be useful. Similarly, depending on the characteristics of the log, some clustering approaches could be more useful than others. These and many more challenges can be found from inspecting the state-of-the-art as represented in our framework and form a good basis for future research.

4 Paper 2: Are We Doing Things Right? An Approach to Measure Process Inefficiencies in the Control Flow

4.1 Abstract

A major dimension for assessing organizational performance is efficiency, i.e., the amount of output obtained from a given input. Organizational efficiency is closely connected to business process efficiency. Inefficiently executed processes may consume a lot of resources and still not achieve their internal goals. Because "you cannot improve what you cannot measure", process mining tools try to quantify process inefficiency with rather basic indicators, which provide only limited information. This paper introduces an approach that measures process inefficiencies in the control flow, taking factors like an activity's intended position in the trace and the allowed number of repetitions into account. Our evaluation results show that the process performance indicators that our approach defines capture aspects of process inefficiency that have not been taken into account in the baseline indicator that is currently provided in process mining tools.

4.2 Introduction

Operational efficiency, i.e., the amount of output obtained from a given input, is one of the major dimensions for assessing organizational performance (Davis & Peri, 2002). The efficiency of an organization is closely connected to the efficiency of its business processes (Melville et al., 2004). Inefficiently executed processes may consume a lot of time, cost, and personnel resources and still not achieve their internal goals. This is particularly problematic for support processes, like HR or purchasing, whose goal is to enable the execution of the organization's value-creating core processes.

For successful business process management (BPM), measuring process inefficiency is the first step towards improvement (Van Looy & Shafagatova, 2016). Inefficiencies in process execution like rework or change activities, loops, or cancellations can be observed in the event log that captures the execution of the respective process in an IT system. Companies like Uber (El-Wafi, 2020) and Siemens (Rowlson, 2020) have been trying to harmonize their processes and reduce rework and loops to achieve higher efficiency levels. The identification of efficient processes also is a challenge for robotic process automation (RPA) because automating inefficient processes will amplify the inefficiency (Reinkemeyer, 2020). However, there is no explicit indicator for process inefficiency in the process mining literature (Van Looy & Shafagatova, 2016).

Process mining vendors appear to have recognized an industrial need for measuring process inefficiencies (Aull, 2020), but the indicators that they provide tend to be very basic and therefore do not provide a lot of value. For example, in Celonis, a case is labelled as inefficient if it contains an inefficient activity (e.g., change price) (Badakhshan et al., 2020). Process inefficiency is then defined as the ratio of inefficient cases in an event log. This indicator misrepresents process inefficiency with regard to the above understanding of efficiency as a relation of input and output, because it does not consider cases that include more than one inefficient activity.

In this paper, we present a novel approach for measuring process inefficiencies (AMPI). It is based on the idea that a process inefficiency is caused by either the type of activity (e.g., a deletion), its location (the activity is executed at the wrong position in the trace), or its frequency (the activity is executed more often than intended). Our approach defines a set of performance indicators, which measure the inefficiency of individual traces independent from the overall event log. It is defined on the trace level, hence allowing for a comparison between cases, and relies only on the control flow, hence requiring only a partial order within the trace. Therefore, we report on related work in Section 4.3. The process of developing AMPI is described in Section 4.4 and evaluated in Section 4.5. Section 4.6 discusses contributions and limitations, before the paper is concluded in Section 4.7.

4.3 Related Work

Our research takes a process-centric view on organizational performance, an important construct in strategic management. Researchers have defined countless measures for assessing an organization's performance in so-called performance measurement models (PMMs) (Van Looy & Shafagatova, 2016). Organizational PMMs (Cross & Lynch, 1988; Kaplan & Norton, 2001) cover all aspects of the business, whereas business process PMMs (Kueng, 2000; Neely et al., 2000) focus on individual business processes, which makes them particularly relevant for BPM (Van Looy & Shafagatova, 2016).

Business processes are accepted as a significant construct in all of the mentioned PMMs. The Balanced Scorecard offers four main perspectives to managers (customers, internal processes, innovation, improvement activities). To define and measure the internal process perspective, companies must consider different variables for their business processes, e.g., project closeout cycle, project performance effectiveness index, and rework (Kaplan & Norton, 1998). Similarly, the 4-level pyramid model by Cross and Lynch (Cross & Lynch, 1988) contains "the vision" on the top and "operational measures" (including quality, delivery, process time, and cost) on the bottom.

Process performance measurement systems (PPMSs) play a major role in improving business processes for any process-oriented organization (Kueng, 2000). Dumas et al. describe time, cost, quality, and flexibility as the main dimensions of a PPMS (2018). Because those dimensions are multi-faceted and rather abstract, various performance indicators are suggested to quantify the goals associated with each dimension. Such indicators for general organizational performance are called key performance indicators (KPIs). Process performance indicators (PPIs) are the process-related version of KPIs (Rosenberg et al., 2011).

They should satisfy the SMART (specific, measurable, achievable, relevant, time-bounded) criteria for KPIs (Shahin & Mahbod, 2007) and also be expressive, understandable, traceable, and automatically measurable (del-Río-Ortega et al., 2013). The PPINOT metamodel allows for an unambiguous and complete definition and implementation of PPIs (del-Río-Ortega et al., 2013) and is enhanced by a graphical notation for defining and visualizing PPIs with business process models (del-Río-Ortega et al., 2019).

Process inefficiency, which is defined as the performance gap in comparison to a best practice, can either be design-related or intrinsic, i.e., related to process execution (Burger & Moormann, 2009). Design-related inefficiencies are highly domain-specific and have to be addressed during the design stage of the business process life cycle (Dumas et al., 2018). However, even an efficiently designed process needs to account for the necessary flexibility in execution (B. Hompes et al., 2016), so the mere conformance of an execution is not an indicator for its organizational performance (Van Den Ingh et al., 2020). For example, a process-executing IT system must allow for the termination of an incomplete process instance (e.g., on cancellation) or give employees the opportunity to correct or update wrong or outdated data. Nevertheless, those activities should be avoided to reduce intrinsic inefficiencies and improve the process's performance.

A recent review on PPIs did not find any explicit indicator for process inefficiency in the process mining literature (Van Looy & Shafagatova, 2016). However, there are a few studies that have dealt with measuring process inefficiencies. Dohmen and Moormann apply a three-stage approach to discover the association of intrinsic process execution characteristics and their efficiency score (Dohmen & Moormann, 2010). This case study measures the inefficiency of banking transactions by comparing them to the best-practice transactions. This means that the measure cannot be assessed for individual cases, but always depends on the most efficient transaction in the log. Also, the proposed method is domain-specific and limited to the financial sector. Van Den Ingh et al. describe an approach to measure process performance based on process mining (Van Den Ingh et al., 2020). Their approach evaluates variants of a P2P process based on control flow and context, but control flow inefficiency is assessed with very basic indicators, e.g., percentage of activities that was executed more than once. Höhenberger &

Delfmann (2015) applied automated model query approaches to collect weakness (process model) patterns. Using their discovered patterns, they were able to find weaknesses in new process models from different context. This approach emphasizes the detection of patterns as the first step to discover and solve underlying root causes.

4.4 AMPI: An Approach for Measuring Process Inefficiency

4.4.1 Objectives

In the following, we describe AMPI, which, once applied to a specific process in a concrete context, results in a set of inefficiency PPIs. AMPI targets the following types of inefficiencies.

4.4.1.1 Non value-adding activities Activities can add value to the process by providing value to the customer (e.g., the receipt of goods in a P2P process) or to the business (e.g., the approval of a payment) (Dumas et al., 2018). If activities do not fall within those two categories, they can be seen as inefficient. This includes activities like hand-offs or cancellations, but also so-called rework activities, like price changes (e.g., El-Wafi (2020)), which revoke the outcome of a previously executed activity.

4.4.1.2 Loops Loops indicate a repetition of activities, which is perceived as inefficient because the same work has to be done twice. This is not the case for all activities. In a P2P process, for example, more than one occurrence of “release purchase requisition item” per case would be inefficient (El-Wafi, 2020, p. 82), but a case can have multiple goods receipts. For unintended loops, the magnitude of the inefficiency is determined by number of iterations (the more executions, the more inefficient) and their length. Research has provided evidence for a “bullwhip effect” at a process level, meaning that occurring mistakes or problems should be addressed and handled as soon as possible, because otherwise, its consequences become more damaging (Mahendrawathi et al., 2018). If, for example the price of a purchase order item is changed twice in a row, it is less inefficient than if the second change happens after the invoice is confirmed.

4.4.1.3 Wrong start and end activities Start and end activities of a process are particularly relevant, because they provide the basic information for initiating or concluding a case. If start activities are wrongly positioned, the other activities will lack information and therefore be incomplete. If end activities are wrongly positioned, follow-up activities cannot be executed efficiently. A purchase order item, for example, should not be created at the end of a P2P process, because it provides the informational basis for the entire case.

4.4.2 Outline

These examples show that for an insightful measure of process inefficiency, it is not sufficient to count the ratio of inefficient activities. Instead, AMPI takes the following steps to calculate the ratio of inefficient behavior in a trace.

1. Defining Activity Clusters: The basis of any inefficiency lies in the nature of the activities. We define a generic framework for classifying activities by their intended frequency (0, 1, multiple) and location (start, core, end). This results in nine activity clusters, which form the basis of AMPI.

2. Identifying Drivers of Inefficient Behavior: Some inefficiencies are revealed or exacerbated by certain conditions in the control flow, like repetitions of the same activity. We call those conditions "drivers" of inefficient behavior. They are defined and computed for each occurrence of an activity in a trace.

3. Calculating Cluster-based Inefficiencies: We calculate the (absolute) inefficiency per cluster by aggregating the relevant drivers of inefficiency for all activities in the cluster and all occurrences of that activity.

4. Computing the Trace-level Inefficiency: To compute the trace-level inefficiency, we normalize the cluster-based inefficiencies and then take the average across all clusters.

In the following, we say that $A = \bigcup_{j=1}^{m} a_j$ is the *universe of activities.* An *occurrence* or *event* $e(a)$ denotes the execution of an activity a within a process and can be denoted as a, if the context is clear. A *case* denotes one execution of a process, which has at least two attributes (an ID and a trace). A *trace* t is a finite sequence of occurrences $e_1, ..., e_n$, where each $t[k] = e_k$ denotes an occurrence of an activity $a \in A$ at the kth position of t, and $|t| = n$ denotes the *length* of the trace. $A^t = \bigcup_{e(a) \in t} a$ is the set of distinct activities for t. The *frequency* of any distinct activity $a^t \in A^t$ is defined as the number of occurrences, written $|a^t|$. Occurrences of a in t are denoted as a_i^t, where a_1^t indicates the first occurrence of a in t and a_l^t, $1 \leq l \leq |t|$ indicates the last occurrence of a in t. $[a_i^t]$ denotes the *location* of an occurrence in a trace, i.e., the index k, for which $e_k = a_i^t$.

4.4.3 Assigning Activities to Clusters

The first dimension of process inefficiency is the intended frequency of activities, which has three categories. The red category contains undesired activities, which do not add any value and should therefore not occur at all. The green category contains desired activities, which can be executed multiple times. Activities that should explicitly occur only once per trace fall into the yellow category.

The second dimension of process inefficiency is an activity's intended location in the trace. Therefore, we partition the trace into three sections. The start section and the end section contain the first and last activities of a trace, respectively. The core of the trace comprises all remaining activities in between the start and end. This separation recognizes the crucial role that start and end activities play in the trace, but leaves enough flexibility for process execution.

Table 8: Activity clusters defined by intended activity frequency and location

		Location		
		Start	Core	End
Frequency	0	Cluster S0	Cluster C0	Cluster E0
	1	Cluster S1	Cluster C1	Cluster E1
	Multiple	Cluster S2	Cluster C2	Cluster E2

Combining the two dimensions with three categories each results in nine distinct activity clusters, as listed in Table 8. Readers should note that the same activity will usually be part of multiple clusters. The activity clusters are based on theoretical considerations and therefore independent from the analyzed process itself. To apply it to a concrete process, we need e.g., a process analyst or a domain expert to assign process activities to clusters.

4.4.4 Identifying Drivers of Inefficient Behavior

The clusters impose an intended behavior on their activities. Inefficiencies occur if activities do not adhere to this behavior. To identify these inefficiencies, we need to inspect the control flow context in which activities occur. This context reveals some inefficiencies, e,g., repetitions of yellow activities, and exacerbates others, e.g,, loops. Below, we discuss four drivers of inefficient behavior in the control flow. Each driver is defined for an activity or an occurrence of an activity. Because AMPI should allow for the comparison of individual traces, the drivers are normalized to the trace length. When computing the overall trace inefficiency, we can then sum up the normalized drivers and obtain a normalized inefficiency value. This normalization at driver level is necessary, because the normalization basis depends on the nature of the cluster. Therefore, the following equations are only valid for their respective cluster and its assigned activities.

4.4.4.1 Repetition The intended activity frequency determines whether the repetition of an activity adds additional value to the process. Any additional occurrence above what is intended by the respective cluster is regarded as inefficient. For most clusters, those occurrences can just be counted when computing the overall trace inefficiency, but for clusters S1 or E1, which cover singular start or end activities, exceeding the intended frequency is particularly damaging. We account for such violations by individually factoring in their repetitions. Moreover, we also count for the variation of inefficient activities within one case, i.e., “multiple occurrences of one inefficient activity” is better (less inefficient) than “multiple occurrence of multiple inefficient activities”. The rationale behind this decision lays in the nature of organizations where involving different actions usually requires extra communication and handover costs. However, performing the same action in the same company most likely costs relatively less overhead.

$$r_{S1}(a_i^t) = r_{E1}(a_i^t) = \frac{i-1}{\max(1, |t|-1)} \tag{1}$$

4.4.4.2 Location The location of an activity is defined as its index in the trace. For each activity, the intended location can either be 0 (start activity), $|t|-1$ (end activity), or any number in between (core activity). We assume inefficient behavior to be less severe early in the trace, so we differentiate between the first and all other occurrences. For C0 activities, an inefficiency of zero cannot be achieved if any C0 activity occurs at the core of the trace. For C1 activities, we assess their locations with help of two measures, l_{C1} and b_{C1}, to avoid twofold evaluation of activities where $l_{C1} = l_{S1}$. Whereas l_{C1} evaluates the location of C1 activities at the core, b_{C1} only assesses the start and end of the trace in terms of C1 activities. Location measures for clusters S0, E0, and C2 are not specified here, because the (absolute) location is the only relevant driver for these clusters.

$$l_{C0}(a_i^t) = \begin{cases} \frac{[a_i^t]}{\max(1,|t|-2)}, & \text{for } i = 1 \\ \\ \frac{[a_i^t]-1}{\max(1,|t|-3)}, & \text{otherwise} \end{cases} \tag{2}$$

$$b_{C1}(a^t) = \begin{cases} 1, & \text{if } t[0] \text{ and } t[|t|-1] \in C1 \\ 0.5, & \text{if } t[0] \text{ or } t[|t|-1] \in C1 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

$$l_{S1}(a_i^t) = \frac{[a_i^t]}{\max(1, |t|-1)} \tag{4}$$

$$l_{S2}(a_i^t) = \frac{1 - i + [a_i^t]}{\max(1, |t|-i)} \tag{5}$$

$$l_{E1}(a_i^t) = 1 - \frac{[a_i^t]}{\max(1, |t|-1)} \tag{6}$$

$$l_{E2}(a_i^t) = 1 - \frac{1 - i + [a_i^t]}{\max(1, |t|-i)} \tag{7}$$

4.4.4.3 Distance Not all repetitions of red or yellow activities are equally inefficient. To assess the impact of a repeated activity on the trace, we measure the distance between activities, defined as the absolute difference between their locations. We recall our assumption that repetitions are less inefficient if they are close together and ensure that we always compute the distance of a repetition to the first occurrence of the activity. As yellow activities should only occur once at most, any distance of a repeated activity is treated inefficient. In contrast, the optimal distance for green activities highly depends on the number of repeated activities. Note that $d_{C1} = d_{E1} = d_{S1}$, and $d_{E2} = d_{S2}$.

$$d_{C0}(a_i^t) = \frac{[a_i^t] - [a_1^t] - 1}{\max(1, |t|-3)} \tag{8}$$

$$d_{S1}(a_i^t) = \frac{[a_i^t] - [a_1^t]}{\max(1, |t| - 1)} \tag{9}$$

$$d_{S2}(a_i^t) = \frac{1 - i + [a_i^t] - [a_1^t]}{\max(1, |t| - i)} \tag{10}$$

4.4.4.4 Distinct activities In some situations, the previous measures are not sufficient to accurately assess process inefficiency. Consider a trace with a C0 activity at the core. The existence of a second distinct C0 activity at the core impacts the level of inefficiency to another degree than an additional C0-activity of the same type. This must be considered by an additional driver. For this purpose, we define the distinct activities per cluster as the number of distinct activities in a trace that are assigned to the same cluster.

$$da_{C0}(a^t) = \frac{|A^t| - 1}{\max(1, |t| - 3)} \tag{11}$$

$$da_{S2}(a^t) = da_{E2}(a^t) = \frac{|A^t| - 1}{\max(1, |t| - 1)} \tag{12}$$

4.4.5 Calculating Cluster-based Inefficiencies

The drivers of inefficiency are defined on activities or occurrences and can be used to compute the (absolute) inefficiency level of the individual clusters (with regard to one trace). Because the idea behind the start and end dimensions are very similar, the computation and the formulas of the inefficiency levels of clusters S2, S1, and S0 are in line with those of E2, E1, and E0. For this purpose, we analyze the clusters with the same intended frequency together. Related activities then only differ in their intended location.

Activities from S2 may occur multiple times at the start of a trace, so their occurrences do not represent inefficient behavior if they directly follow each other. Hence, repetitions do not impact the inefficiency, but location, distance, and distinct activities must be considered. The optimal location and distance of these activities depend on their number of occurrences. The second occurrence should, e.g., occur at location 1 with a distance of 1 to the first occurrence. The reverse holds true for activities assigned to cluster E2. In contrast, activities from S1 or E1 may occur only once in a trace, so repetitions play a crucial role. Their optimal location is at the start or end of the trace, so it is independent from other activities. The distance between the occurrences is considered, too.

$$\text{absolute inefficiency}_{S2}(t) = \sum_{a^t \in A^t} \sum_{i=1}^{|a^t|} l_{S2}(a_i^t) \times (1 + d_{S2}(a_i^t) + da_{S2}(a^t)) \tag{13}$$

$$\text{absolute inefficiency}_{E2}(t) = \sum_{a^t \in A^t} \sum_{i=1}^{|a^t|} l_{E2}(a_i^t) \times (1 + d_{E2}(a_i^t) + da_{E2}(a^t)) \tag{14}$$

$$\text{absolute inefficiency}_{S1}(t) = \sum_{a^t \in A^t} \sum_{i=1}^{|a^t|} l_{S1}(a_i^t) \times (1 + d_{S1}(a_i^t) + r_{S1}(a_i^t)) \tag{15}$$

$$\text{absolute inefficiency}_{E1}(t) = \sum_{a^t \in A^t} \sum_{i=1}^{|a^t|} l_{E1}(a_i^t) \times (1 + d_{E1}(a_i^t) + r_{E1}(a_i^t)) \tag{16}$$

The absolute inefficiency of S0 and E0 is determined in a different way. Here, most drivers do not have to be assessed because assigned activities should simply never occur. This behavior can be depicted by a binary location variable. In case of cluster S0, this variable is equal to one if the start of a trace is occupied by an assigned activity. The same applies to cluster E0 for the end of a trace.

$$\text{absolute inefficiency}_{S0}(t) = \begin{cases} 1, & \text{if } t[0] \in S0 \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

$$\text{absolute inefficiency}_{E0}(t) = \begin{cases} 1, & \text{if } t[|t|-1] \in E0 \\ \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

The core of a trace contains all activities that occur between the start and end. The activities assigned to clusters C2, C1, and C0 differ in their intended frequency. For the activities in C2, no restrictions are imposed, given that they occur in the core. So, the inefficiency level of cluster C2 is only determined by the location. If those activities are located at the start and end of the trace, the inefficiency level is equal to one. If either the start or the end is occupied by a C2 activity, the level is equal to 0.5 and only if both locations are free of C2 activities, no inefficiency in terms of cluster C2 is present.

The start and end of a trace must also be analyzed for cluster C1. Here, this is done by the bounds measure b_{C1}. In contrast to C2, additional drivers must be considered if a specific C1 activity occurs more than once. In case of repetitions, we assume the level of inefficiency to be moderate if the activities follow each other early in the trace. Based on this assumption, location and distance are relevant for the computation of the inefficiency of cluster C1.

For the analysis of cluster C0, we limit our view on the core of the trace. As those activities should not be executed, any occurrence increases the inefficiency. If a C0 activity occurs nevertheless, it should happen early in the trace, such that it has less impact on the trace overall. This is reflected by the location and distance measures. It follows that in case of repetitions, the optimal distance should be minimized. The more distinct C0 activities there are at the core of the trace, the more activities violate their intended behavior. Consequently, the number of distinct C0 activities increases the inefficiency level of cluster C0.

$$\text{absolute inefficiency}_{C2}(t) = \begin{cases} 1, & \text{if } t[0] \text{ and } t[|t|-1] \in C2 \\ 0.5, & \text{if } t[0] \text{ or } t[|t|-1] \in C2 \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

$$\text{absolute inefficiency}_{C1}(t) = \sum_{a^t \in A^t} (b_{C1}(a^t) + \sum_{i=2}^{|a^t|} (d_{C1}(a_i^t) + l_{C1}(a_i^t))) \tag{20}$$

$$\text{absolute inefficiency}_{C0}(t) = \sum_{a^t \in A^t} \sum_{i=1}^{|a^t|} (1 + da_{C0}(a^t)) \times (d_{C0}(a_i^t) + l_{C0}(a_i^t)) \tag{21}$$

4.4.6 Computing the Trace-Level Inefficiency

Once we have the (absolute) inefficiency values for the clusters, we compute the trace-level inefficiency as follows:

1. Per trace and cluster, we create an artificial worst trace of same length, which contains as much inefficient behavior (in terms of this cluster) as possible. This trace is an auxiliary construct and may not necessarily appear in the event log. It serves as a boundary for 100% inefficiency, such that we can express trace-level inefficiency as a normalized value between 0 and 1.
2. To compute the normalized inefficiency level of a cluster, we compute the absolute inefficiency value of the artificial worst trace (with regard to the respective cluster) according to our definitions above. We obtain the inefficiency level of our trace in question by dividing its absolute inefficiency values by those of the artificial worst trace.
3. The inefficiency of the trace is computed as the average inefficiency level of all clusters. If one cluster is not assigned any activities, this cluster is not considered in the calculation.

Partial PPIs (e.g., for only one activity category) are computed as the average between the respective clusters.

4.5 Evaluation

4.5.1 Outline and Reproducibility

AMPI is evaluated in an experimental analysis by applying it in two different domains and implementing the resulting PPIs.[1] This demonstrates its applicability and generalization in two different contexts. To avoid representational bias, we did not apply any filter (e.g., for unfinished cases) to the logs.

4.5.1.1 Purchasing process (BPI 2019) This event log describes a P2P process with 1.5 million events in 251,734 purchase order items (cases) (Dongen, 2019). Depending on the purchased item, those cases follow four separate flows of activities, which we call item categories 1 to 4. As the optimal control flow and therefore the cluster assignments differ among these categories, we partition the log into four sublogs and assess the inefficiency of each sublog individually.

4.5.1.2 Administrative process (BPI 2020) The second log describes a travel reimbursement process, which distinguishes between domestic and international trips (Dongen, 2020). To account for the different process variants, the data is split into five sublogs: requests for payment (6,886 cases), domestic declarations (10,500 cases), prepaid travel cost (2,099 cases), international declarations (6,449 cases), and travel permits (7,065 cases). Again, we assess each sub log individually.

In the following, we show the separate inefficiency indicators for the start, core, and end categories of each log, defined as the arithmetic mean between the respective clusters. We also

[1]The implementation and the full lists of activity assignments can be found at https://bit.ly/3IS4NPa

Table 9: Inefficiency levels computed by AMPI for BPI 2019

Item Category	Inefficient Cases	PPI: Start			PPI: Core			PPI: End			PPI: All		
		Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
1	0.66	0.0	0.17	0.0	0.0	0.50	0.09	0.0	0.60	0.05	0.0	0.28	0.05
2	0.39	0.0	0.56	0.01	0.0	0.50	0.02	0.0	0.59	0.03	0.0	0.43	0.02
3	1.00	0.0	0.60	0.17	0.0	0.33	0.09	0.0	0.59	0.28	0.1	0.43	0.18
4	0.31	0.0	0.17	0.01	0.0	0.50	0.03	0.0	0.83	0.06	0.0	0.38	0.03
Total	0.41	0.0	0.60	0.01	0.0	0.50	0.02	0.0	0.83	0.04	0.0	0.43	0.03

show the overall log inefficiency, defined as the arithmetic mean of the three. This separation should provide a more detailed insights into the sensitivity of AMPI.

4.5.2 Inefficiencies in a P2P process: BPI 2019

4.5.2.1 Activity Assignment Because this process contains four different variants, we need four different assignments of activities to clusters. Based on the process description, we assume that *Record goods receipt* and *Record subsequent invoice* may occur multiple times for item category 1 (C2). *Record goods receipt* is also assigned to S0 to account for maverick buying. *Create purchase requisition item* is a start activity that may not occur anywhere else in the trace (S1, C0, E0). *Clear invoice* is the intended end activity (E2). These assignments are applied to all other item categories, but have to be adapted to account for the intended process flow. For example, for item category 2, invoices have to be blocked until goods receipt, so *Remove payment block* is assigned to C1.

4.5.2.2 Inefficiency Assessment Table 9 shows the results of computing the inefficiency of the BPI 2019 log. In total, 41% of cases contain some inefficient behavior. Item category 4 contains the least inefficient behavior and item category 2 also contains only 39% inefficient cases. In contrast, all cases from item category 3 are to some degree inefficient. Cases with the highest level of inefficiency often include an unusually high number of change activities, given their trace lengths. For example, the worst cases in terms of the inefficiency at the start begin with changing the approval three times before creating the PO item. The overall worst cases with an inefficiency of 0.83 at the end of the trace belong to item category 4. Such cases should end with the activity *Record goods receipt*, but often continue with change and deletion activities.

4.5.3 Inefficiencies in an administrative process: BPI 2020

4.5.3.1 Activity Assignment This process also contains different flows of activities across the five separate logs. For example, employees require a permission to go on an international trip, so *Permit submitted by employee* is assigned to cluster S1. For a domestic trip, employees can start the process by submitting a declaration instead. Any filing of a document (a permission, a declaration, etc.) can have a positive or negative outcome. For positive outcomes, the end activity is *Payment Handled* is assigned to cluster E1 for all logs. For negative outcomes, the university rejects the corresponding document and the employee needs to accept the rejection, so activities like *Declaration rejected by employee* or *Permit rejected by employee* are assigned to cluster E1.

4.5.3.2 Inefficiency Assessment Table 10 reveals a substantial difference among the two types of trips. Whereas only 10% of domestic trips contain some inefficient behavior, we find 40% inefficient cases for international trips. On average, the inefficiency level is higher for international trips across all sections of a trace. Employees require a permission for those trips, but they often start the process without permission or by filing a declaration instead, which causes the high inefficiency level for the process start.

On average, the end section of a trace exhibits a rather small level of inefficiency, because most cases have a positive outcome (employees get their costs reimbursed in 90.08% of cases).

The trace core does not contain a high level of inefficiency either. This is due to the assignment of activities. Here, we allowed for repetitions of many activities, such that rejections are not necessarily considered inefficient.

Table 10: Inefficiency levels computed by AMPI for BPI 2020

Log	Inefficient Cases	PPI: Start			PPI: Core			PPI: End			PPI: All		
		Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
Dom.	0.10	0.0	0.15	0.00	0.0	0.17	0.00	0.0	0.10	0.00	0.0	0.09	0.00
Int.	0.40	0.0	0.53	0.08	0.0	0.17	0.01	0.0	0.07	0.01	0.0	0.25	0.03
Perm.	0.60	0.0	0.54	0.05	0.0	0.22	0.03	0.0	0.50	0.00	0.0	0.24	0.03
Prep.	0.24	0.0	0.50	0.06	0.0	0.33	0.02	0.0	0.07	0.00	0.0	0.28	0.03
Req.	0.09	0.0	0.14	0.00	0.0	0.17	0.00	0.0	0.54	0.00	0.0	0.20	0.00

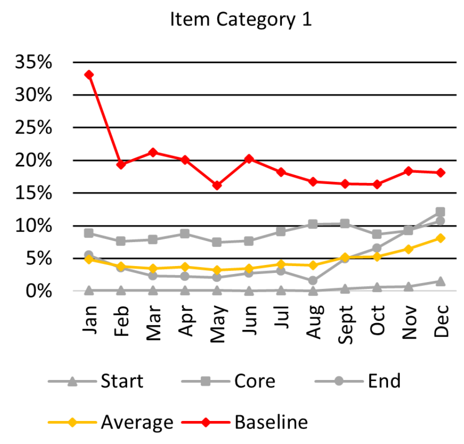
4.5.4 Comparative Evaluation

In addition, we compare the results of AMPI to those of a baseline indicator (BI) that is currently used in a few process mining tools (e.g., Celonis, minit, disco, myInvenio). Like AMPI, BI evaluates only the control flow of a process. It is defined as a binary measure on case or activity level. A case is labeled as inefficient if it contains at least one inefficient activity (e.g., El-Wafi (2020)). Which activities are considered inefficient depends on the context of the process.

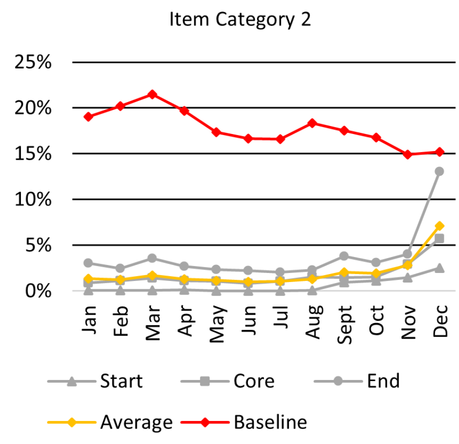
To compare the results of AMPI and BI for BPI 2019, we differentiate between the four item categories and plot the monthly start, core, end, and average inefficiencies, shown in Figure 3.

We see that BI most often calculates a higher inefficiency value than AMPI. BI also tends to decrease towards the end of the year, whereas AMPI indicates an increase across most sections. This increase is particularly strong for the end category and can be explained by unfinished cases, which we did not filter out, to avoid representational bias towards AMPI. For a practical application of AMPI, we suggest using a sliding window approach to avoid a sharp cutoff of unfinished cases.

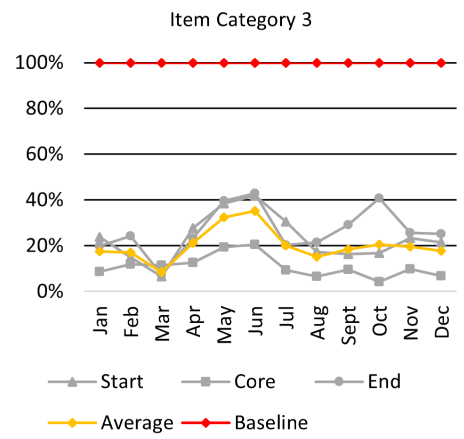
Because BI is a binary measure, whereas AMPI computes a value between 0 and 1, it is more reasonable to evaluate the differences over time instead of the differences between the values. For example, the inefficiency values for item categories 2 and 4 are rather stable according to AMPI, but varies significantly throughout the year according to BI.



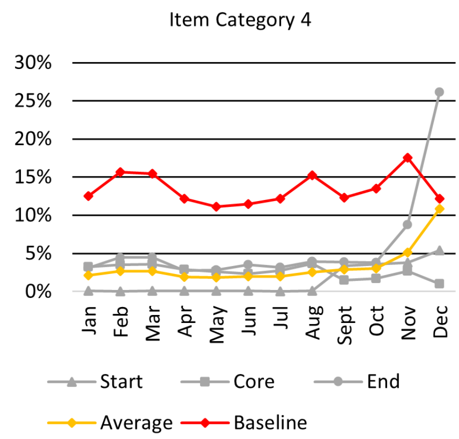
(a)



(b)



(c)



(d)

Figure 3: Comparing AMPI and BI for BPI 2019

The differences between the two approaches also become apparent when analyzing cases of item category 3. Here, BI labels all cases as inefficient. We recall that AMPI also considers all cases of item category 3 to be inefficient, but the degree of inefficiency varies considerably [2].

4.6 Discussion

The comparisons between the inefficiency measures by AMPI and by BI illustrate the contributions of our approach. BI only considers the presence of (presumably) inefficient activities, instead of their order or frequency. Because AMPI considers more drivers of inefficient behav-

[2]Figure B in the Appendix section presents the results for BPI 2020 dataset.

ior, such as activity locations, it is able to find more inefficient cases than BI. Those advantages become more apparent, when comparing the different values over time.

One commonality between AMPI and BI is that both rely on domain or process expertise to identify the activities that contribute to inefficient behavior. Whereas BI only categorizes activities as either efficient or inefficient, AMPI also takes their frequency and location into account to provide a more elaborated view on control-flow inefficiencies. All in all, the differences between the two approaches are caused by their respective nature. BI is a binary measure and evaluates the inefficiency in terms of the presence of inefficient activities. In contrast, AMPI takes a more sensitive approach and evaluates the level of inefficiency across various dimensions.

AMPI still has a number of limitations, both in design and computation. First, it only accounts for inefficiencies that appear in the control flow perspective, although other perspectives (such as time or resource) might have a larger impact on process inefficiency. Second, it recognizes non-value-adding activities, loops, and wrong activity positions, but not other types of inefficiencies such as interdependencies between activities. Third, one could argue that our separation of traces in start, core, and end can overestimate the influence of the start and the end activity and neglects other (core) activities with a potentially higher influence on the overall inefficiency. However, we made this design choice on purpose based on our experiences with real-world data. In defining the clusters, we do not impose a specific location on the core activities, because we have found that this limits the flexibility of the process and that core activities contribute much less to the overall inefficiency. This assessment might change when we apply AMPI to other domains.

The choice of clusters is another important limitation of AMPI. We assume that frequency and location are the two most relevant factors for inefficiency, but there might be others. Our selected categories are also chosen to cover non-value-adding activities, loops, and wrong activity positions, but could be further generalized to define, e.g., a specific intended frequency to each activity. Also, we weigh all activities within one cluster equally, although their actual impact may differ considerably. This could be addressed by defining an individual activity weight, which could be determined by experts or derived from the log itself.

AMPI cannot be calculated for traces with a length smaller than three, because the core of a trace cannot be empty. Also, AMPI only yields meaningful results for finished cases, because unfinished cases are always punished for a wrong end activity. For the overall inefficiency measure (the arithmetic mean between the start, core, and end inefficiency), one could argue that by weighing them all equally, we further increase the overemphasis on start and end activities.

Another limitation concerns our design choice to make AMPI independent of the other traces in the log. This allows us to compute inefficiency levels per trace and compare them across different lengths. However, it also requires the creation of an artificial worst trace as an auxiliary construct to define the "limits" of inefficiency that can hypothetically be achieved. It would be more realistic to use a real-life worst trace as comparison, but this would make the inefficiency of a single trace dependent on the other traces in the log.

The PPIs that follow from applying AMPI fulfill the KPI criteria of being specific (targeted towards process inefficiencies), measurable (shown in the evaluation), achievable (0% inefficiency is possible, although difficult), relevant (shown by the industrial need to measure inefficiency), and time-bounded (inefficiency is only measured for the time span of the event log). They also comply with the PPI requirements (del-Río-Ortega et al., 2013) in terms of traceability and automatic measurement. Their expressiveness and understandability depend on the application context and remain to be evaluated in more practical settings. Yet, these indices are significantly more complex (in terms of calculations, not structure) than common exemplary PPIs (del-Río-Ortega et al., 2013, 2019) and cannot be linked to a few specific concepts (e.g., responsible, informed, scope) due to their context-independent design. Still, the resulting PPIs can be defined and utilized according to organization-specific policies and goals. So, despite this misalignment, AMPI can be framed under the PPINOT metamodel (del-Río-Ortega et al., 2013). Below, we show a partial calculation of the inefficiency PPI for S2. PPINOT is designed as a generic tool to increase clarity of PPI definitions, so complex measures should be explained in natural language.

```
PPI{
    identifier: PPI_S2
    name: control-flow inefficiency cluster S2
    relatedTo: #process name
    goals:  reduce the level of control-flow inefficiency in the
            process with regards to cluster S2
    #sum over all distinct activities i of type S2
    target: simpleTarget.upperBound: 0.00 #desired value of 0.00
    scope: ProcessStateFilter.processState: finished
}
```

4.7 Conclusion

In this paper, we present our novel approach for measuring process inefficiency (AMPI). When applied to a specific process in a concrete context, AMPI yields a set of process performance indicators (PPIs) for measuring process inefficiency. AMPI accounts for several types of inefficient behavior (non-value-adding activities, loops, wrong activity positions) and allows for a comparison between cases. Compared to the baseline indicator for process inefficiency, which is currently used by a major process mining tool, AMPI provides a more sensitive and realistic way of quantifying inefficiencies in the control-flow of a process, which gives process analysts a better chance of finding improvement potentials.

Although AMPI is a significant improvement over the state of the art in measuring process inefficiencies, this problem is far from being solved. In future work, we want to address some of the limitations listed above and extend AMPI to go beyond control flow and include other factors like execution time. This might entail taking an even more domain-specific perspective to be able to identify different types of inefficiencies in processes. Our definition of inefficiency in this study does not reflect the absence or the disarrangement of desired activities. The required input for this study, assigning the clusters, is expected to be done manually, thus leaving space for human errors. This step can be automated in supplementary studies by means of more advanced techniques, e.g., semantic analysis, and deep learning techniques. Also, we want to follow up on a theoretical observation that we made during the evaluation. Our results suggest that the process inefficiency changes rather smoothly over time and does not show dramatic fluctuations. It appears to behave (or change) as an institutionalized habit. Hence, we could use AMPI as a basis to theorize about organizational routines and their reactions to endogenous and exogenous changes (Grisold et al., 2020).

5 Paper 3: Fig4PM: A Library for Calculating Event Log Measures

5.1 Abstract

Calculating event log measures (also known as features, metrics, and characteristics) is a common task required by many process mining applications. Process mining research studies and industrial applications often need to generate measures depending on their requirements. This has resulted in a plethora of event log measures being (re-)invented and (re-)implemented on different platforms. Fig4PM is an attempt toward building a standard, comprehensive, and reusable library for calculating event log measures. The current version of this open-source program offers 73 distinct control-flow measures either directly extracted from the literature (48 measures) or derived from the existing measures (25 measures). Eventually, our objective is to build a standard public Python library to facilitate feature generation in process mining applications.

5.2 Introduction

Process mining projects typically start with extracting data from a process-aware information system and transforming them into an event log (Van der Aalst, 2009). These event logs serve as input for virtually all process mining applications. In order to characterize the event logs and assess the specific differences (and similarities) among the traces, process analysts often employ event log measures, i.e., "numeric representations of raw data" (Zheng & Casari, 2018). These measures can provide a priori insights about a log, which can then be used to draw conclusions about their properties. Typically, a measure is calculated at trace level and then aggregated to represent the event log characteristics. For example, calculating the length of each trace helps building the *average trace length* at event log level.

A wide range of process mining applications utilize such measures. We conducted a literature review to collect the studies that considered implementing new measures based on an event log's control-flow and found 21 scientific papers [3] ranging from 2001 to 2020. Interestingly, we noticed a certain level of overlap among these studies, i.e., different studies do not refer to fully-distinct and exclusive measures. According to the results of our literature review, many approaches require implementing measures, including (but not limited to) data preprocessing (Fani Sani et al., 2019), data quality (Suriadi et al., 2017), predictive process mining (Márquez-Chamorro et al., 2017), approaches that use deep learning techniques (Taymouri, La Rosa, et al., 2020), business process simulation (Martin et al., 2016), process complexity analysis (Augusto et al., 2022), and trace clustering (Zandkarimi et al., 2020).

To avoid the repeated (re-)invention and (re-)implemented of the same event log measures on different platforms, we introduce the Fig4PM library.[4] It provides researchers and practitioners with a basic library to access previously implemented event log measures and is specifically set out to be a starting point for ongoing development efforts. Prospective users may contribute to this project by developing new measures, improving the existing functions, add more data connectors, and improve its overall performance.

5.3 Measures

In Fig4PM, we distinguish two types of measures based on the underlying data structure. Linear measures perceive a trace as an array, matrix, or sequence of letters (a string), whereas non-linear measures perceive a trace as a directed graph, i.e., nodes represent activities while sequences determine edges (Zandkarimi et al., 2020).

[3] Material is available at https://doi.org/10.6084/m9.figshare.14912313.v2.
[4] Code is available at https://github.com/f-zand/fig4pm.

5.3.1 Measures Derived From Linear Structures

Table 5.3.1 lists the measures derived from linear structures that were identified in the literature. For each measure, we list its abbreviation, description, and literature source. The measures are separated into groups based on their literature source and intended purpose.

The first two groups provide a brief overview of the log size and variability. The large third group measures structuredness and variance, i.e., risk of producing a Spaghetti model (Swennen et al., 2015). To quantify these properties, we can measure reoccurring behavior in terms of self-loops and repetitions as well as the number of start and end events which concern variability in initialization or termination. As more elaborate measures for structuredness, we measure the *number of distinct traces per 100 traces* (*tcpht*), *absolute trace coverage* (*tco*) and the *relative trace coverage* (*rtco*). The lower *tcpht*, the more structured the underlying event log. *tco* represents the minimum number of distinct traces required to cover 80% of all traces in the log hence, evaluating the variants' frequencies. Relating *tco* to *ntc* yields the relative trace coverage, which is better suited for comparison across different (sub-)logs.

The fourth group consists of several measures based on density, similarity (diversity), and complexity. The fifth group measures event log entropy using 3 different methods.

Table 5.3.1: Literature-Based Measures - Linear Structure

Abbreviation	**Measure**
ne	Total number of events Ribeiro et al. (2014), Kherbouche et al. (2017), Günther (2009)
nec	Total number of event classes Ribeiro et al. (2014), Kherbouche et al. (2017), Günther (2009)
nt	Total number of traces Ribeiro et al. (2014), Kherbouche et al. (2017), Günther (2009)
ntc	Total number of trace classes Ribeiro et al. (2014), Kherbouche et al. (2017)
atl	Average trace length Swennen et al. (2015), Van den Broucke et al. (2014)
mitl	Minimum trace length Swennen et al. (2015), Van den Broucke et al. (2014)

Abbreviation	Measure
matl	Maximum trace length Swennen et al. (2015), Van den Broucke et al. (2014)
ats	Avg. trace size (level of detail) Benner-Wickner et al. (2014), Kherbouche et al. (2017), Günther (2009)
nsec	Number of distinct start events Ribeiro et al. (2014), Swennen et al. (2015)
ntec	Number of distinct end events Ribeiro et al. (2014), Swennen et al. (2015)
ntsl	Abs. number of traces with a self-loop Swennen et al. (2015)
ntr	Abs. number of traces with a repetition Swennen et al. (2015)
rnsec	Rel.number of distinct start events Swennen et al. (2015)
rntec	Rel. number of distinct end events Swennen et al. (2015)
rntsl	Rel. number of traces with a self-loop Swennen et al. (2015)
rntr	Rel. number of traces with a repetition Swennen et al. (2015)
anslt	Avg. number of self-loops per trace Swennen et al. (2015)
manslt	Max. number of self-loops per trace Swennen et al. (2015)
asslt	Avg. size of self-loops per trace Swennen et al. (2015)
masslt	Max. size of self-loops per trace Swennen et al. (2015)
tcpht	Number of distinct traces per hundred traces Swennen et al. (2015)
tco	Absolute trace coverage Swennen et al. (2015)
rtco	Relative trace coverage Swennen et al. (2015)
edn	Event density Kherbouche et al. (2017), Benner-Wickner et al. (2014)
thr	Traces heterogeneity rate Kherbouche et al. (2017)
tsr	Trace similarity rate Kherbouche et al. (2017)
cf	Complexity factor Kherbouche et al. (2017)
std	Simple trace diversity Benner-Wickner et al. (2014)
atd	Advanced trace diversity Benner-Wickner et al. (2014)
tentr	Trace entropy C. Back et al. (2018)
prentr	Prefix entropy C. Back et al. (2018)
abentr	All-block entropy C. Back et al. (2018)

5.3.2 Measures Derived From Non-Linear Structure

Table 5.3.2 lists the literature-based measures derived from non-linear structures. In comparison to the linear measures, their number is rather limited. Many measures from the literature require post-discovery knowledge which is out of scope for this study. The remaining measures mainly focus on the directly-follows-graph (DFG) of the event log, quantifying the relationship between its nodes N and edges A.

Table 5.3.2: Literature-Based Measures - Non-Linear Structure

Abbreviation	**Measure**
N	Number of nodes / vertices
A	Number of arcs / edges
$gcnc$	Coefficient of network connectivity Mendling (2007), Latva-Koivisto (2001)
$gand$	Average node degree Mendling (2007)
$gmnd$	Maximum node degree Mendling (2007)
gdn	Density Mendling (2007)
gst	Structure Günther (2009)
gcn	Cyclomatic number Latva-Koivisto (2001)
gdm	Graph diameter Mendling (2007)
gcv	Number of cut vertices Van den Broucke et al. (2014)
$gsepr$	Separability ratio Mendling (2007)
$gseqr$	Sequentiality ratio Mendling (2007)
gcy	Cyclicity Mendling (2007)
gaf	Affinity Günther (2009)
$gspc$	Simple path complexity Pentland, Liu, et al. (2020)

5.3.3 Self-Developed Measures

Inspired by the initial set of measures, we created 25 new measures to improve comprehensiveness and cover more topics. Linear structure includes measures focusing on frequency, connectedness, trace length, trace profile, and spatial proximity. Non-linear structure measures include measures based on modularity, cut-vertices, and activity labeling.

Section 5.4 provides a summary of the literature review and a list of all measures plus their respective formulas.

5.4 Supplementary content

5.4.1 Formulas of literature-based measures — Linear structure

Table 5.4.1: Formulas of literature-based measures — Linear structure

Description	Formula
Total number of events	$ne = \sum_{e \in \mathcal{E}} e$
Number of event classes	$nec = \sum_{i=1}^{ne} \sum_{j=i+1}^{ne} (e_i \equiv e_j) \| e_i, e_j \in \mathcal{E}$
Total number of traces	$nt = \sum_{\sigma \in \mathcal{L}} \sigma$
Number of trace classes	$ntc = \sum_{i=1}^{nt} \sum_{j=i+1}^{nt} (\sigma_i \equiv \sigma_j) \| \sigma_i, \sigma_j \in \mathcal{L}$
Average trace length	$atl = \frac{1}{nt} \sum_{i=1}^{nt} \|\sigma\|$
Minimum trace length	$mitl = min(\|\sigma_1\|, ..., \|\sigma_{nt}\|)$
Maximum trace length	$matl = max(\|\sigma_1\|, ..., \|\sigma_{nt}\|)$
Average trace size	$ats = \frac{1}{nt} \sum_{i=1}^{nt} \sum_{j=1}^{\|\sigma\|} \sum_{k=j+1}^{\|\sigma\|} (e_j \equiv e_k \| e_j, e_k \in \sigma)$
Number of distinct start events	$nsec = \|sec\|$ with sec being the set of all start events occurring in event log $\mathcal{L}$
Number of distinct end events	$ntec = \|tec\|$ with tec being the set of all end events occurring in event log $\mathcal{L}$
Absolute number of traces with a self-loop	$ntsl = \sum_{\sigma \in \mathcal{L}} \sigma \| sl \subset \sigma, sl \in slsl$
Absolute number of traces with a repetition	$ntr = \sum_{\sigma \in \mathcal{L}} \sigma \| r \subset \sigma, r \in rsl$
Relative number of distinct start events	$rnsec = \frac{nsec}{nec}$

Description	Formula
Relative number of distinct end events	$rntec = \frac{ntec}{nec}$
Average number of self-loops per trace	$anslt = \frac{1}{nt}\sum_{i=1}^{nt} \lvert sls_i \rvert$
Maximum number of self-loops per trace	$manslt = max(\lvert sls_{\sigma_1} \rvert, ..., \lvert sls_{\sigma_{nt}} \rvert)$
Average size of self-loops per trace	$asslt = \frac{1}{ntsl}\sum_{i=1}^{nt}\sum_{j=1}^{sls} \lvert sl \rvert$
Maximum size of self-loops per trace	$masslt = max(max(\lvert sl \rvert \forall sl \in sls_{\sigma_1}), ..., max(\lvert sl \rvert \forall sl \in sls_{\sigma_{nt}}))$
Number of distinct traces per hundred traces	$tcpht = \frac{ntc}{nt} \times 100$
Absolute trace coverage	$tco = \sum_{i=1}^{nt} \sigma_i^{f_i} \; s.t. \sum_{\sigma \in \mathcal{L}} f_\sigma \geq 0.8 \times nt$
Relative trace coverage	$rtco = \frac{tco}{ntc}$
Event density	edn
Traces heterogeneity rate	$thr = \frac{ln(ntc)}{ln(nt)}$
Trace similarity rate	$tsr = \frac{2}{ntc \times (ntc-1)} \times \sum_{i=1}^{ntc}\sum_{j=i+1}^{ntc} \frac{max(\lvert\sigma_i\rvert, \lvert\sigma_j\rvert) - LD(\sigma_i, \sigma_j)}{max(\lvert\sigma_i\rvert, \lvert\sigma_j\rvert)}$
Complexity factor	$cf = (ln(ntc)^{((1-tsr)+edn)} \times ats)$
Simple trace diversity	$std = 1 - \frac{ats}{nec}$
Advanced trace diversity	$atd = \frac{2}{ntc \times (ntc-1) \times atl} \times \sum_{i=1}^{nt}\sum_{j=i+1}^{nt} LD(\sigma_i, \sigma_j)$
Trace entropy	$tentr = -\sum_{i=1}^{n} p_{\sigma_j} log_b p_{\sigma_j}$
Prefix entropy	$tentr = -\sum_{i=1}^{n} p_{pr_j} log_b p_{pr_j}$
All-block entropy	$tentr = -\sum_{i=1}^{n} p_{ab_j} log_b p_{ab_j}$

5.4.2 Formulas of literature-based measures — Non-linear structure

Table 5.4.2: Formulas of literature-based measures — Non-linear structure

Description	Formula
Number of nodes / vertices	$N = \|V(G)\|$
Number of arcs / edges	$A = \|E(G)\|$
Coefficient of network connectivity / complexity	$gcnc = \frac{A}{N}$
Average node degree	$gand = \frac{1}{N}\sum_{i=1}^{N} deg(v)\|v \in V(G)$
Maximum node degree	$gmnd = max(deg(v_1), .., deg(v_N))$
Density	$gdn = \frac{A}{N \times (N-1)}$
Structure	$gst = 1 - \frac{A}{N^2}$
Cyclomatic number	$gcn = A - N + 1$
Graph diameter	$gdm = max_{u,v} d(u,v)$
Number of cut vertices	$gcv = \|v_{cut}\|$
Separability ratio	$gsepr = \frac{gcv}{N}$
Sequentiality ratio	$gseqr = \frac{\|e_{ncn}\|}{A}$
Cyclicity	$gcy = \frac{\|n_{cyc}\|}{N}$
Affinity	$gaf = \frac{2}{nt \times (nt-1)} \times \sum_{i=1}^{nt}\sum_{j=i+1}^{nt} \frac{\|F(\sigma_i) \cap F(\sigma_j)\|}{\|F(\sigma_i) \cup F(\sigma_j)\|}$ with $F(\sigma_i)$ and $F(\sigma_j)$ being the directly following relations of traces σ_i and σ_j
Simple path complexity	$gspc = \|simpa(G)\|$

5.4.3 Literature review on process features

Table 5.4.3: Literature review on process features

Article Title	Reference
Finding a complexity measure for business process models	Latva-Koivisto (2001)
A Discourse on Complexity of Process Models	Cardoso et al. (2006)
Process control-flow complexity metric: An empirical validation	Cardoso (2006)
Understanding the Occurrence of Errors in Process Models based on Metrics	Mendling, Neumann, et al. (2007b)
On the correlation between process model metrics and errors	Mendling, Neumann, et al. (2007a)
What Makes Process Models Understandable?	Mendling, Reijers, et al. (2007)
Detection and Prediction of Errors in EPC Business Process Models	Mendling (2007)
Process mining in flexible environments	Günther (2009)
Prediction of Business Process Model Quality Based on Structural Metrics	Sánchez-González et al. (2010)
Measurement in business processes: a systematic review	González et al. (2010)
Uncovering the Relationship Between Event Log Characteristics and Process Discovery Techniques	Van den Broucke et al. (2014)
Slice, Mine and Dice: Complexity-Aware Automated Discovery of Business Process Models	Ekanayake et al. (2013)
Examining Case Management Demand Using Event Log Complexity Metrics	Benner-Wickner et al. (2014)
A Recommender System for Process Discovery	Ribeiro et al. (2014)
Capturing Process Behavior with Log-Based Process Metrics	Swennen et al. (2015)
Towards a better assessment of event logs quality	Kherbouche et al. (2017)
Detecting Concept Drift in Processes using Graph Metrics on Process Graphs	Seeliger et al. (2017)
Towards an Entropy-Based Analysis of Log Varisability	C. Back et al. (2018)
Case notion discovery and recommendation: automated event log building on databases	Murillas et al. (2020)
Entropy as a Measure of Log Variability	C. O. Back et al. (2019)
The Dynamics of Drift in Digitized Processes	Pentland, Liu, et al. (2020)

6 Paper 4: How Errors Shape Production Routines: An Empirical Examination of Digital Traces in Manufacturing

6.1 Abstract

Prior literature notes that errors can be an important source of reflection and learning for employees. However, errors are often perceived as uncomfortable experiences, with unclear implications for employees' subsequent performance. Using a novel longitudinal data set of 75K reported errors associated with 3M production batches over six years at a large Dutch public holding company, we investigate the effect of reporting errors on the performance of organizational routines (i.e., interdependent patterns of action). Based on prior research, we theorize that errors are associated with post-error improvement in accuracy (PIA) measured by reporting accuracy and post-error slowing (PES) measured by throughput time. Our results show that reporting accuracy increases by 0.4 percentage points while throughput time, surprisingly, decreases by 7 percent for batches started on the day after an error has been reported. We find that performance only changes after an error is reported and not when it actually occurs. Interestingly, more severe errors are associated with higher same-day reporting accuracy, but not with next-day reporting accuracy, and the relationship between errors and performance becomes stronger for routines with few prior complaints. Our results are robust to various endogeneity checks using instrumental variables and a difference-in-differences analysis. We provide managerial implications for designing effective error management policies.

6.2 Introduction

The digital transformation has wide-ranging implications for the use of information technology (IT) to produce and collect data on organizational processes. This creates numerous opportunities to study how organizations function (Pentland et al., 2021; Turner & Rindova, 2018). Research on organizational routines, commonly defined as "generative systems that produce repetitive, recognizable patterns of interdependent action carried out by multiple participants" (Pentland & Feldman, 2008, p. 236), can especially benefit from the fact that IT creates large amounts of real-time data that give granular insights into how processes are executed. To trace processes in organizations through data, a recent research stream proposes using digital traces, i.e., time-stamped event log data typically used for process mining (Berente et al., 2019; Dumas et al., 2018; Van der Aalst, 2013). Although researchers increasingly acknowledge the usefulness of trace data to study organizational processes at a more granular level (Giardili et al., 2022; Pentland, Recker, et al., 2020; Pentland et al., 2021), little research thus far uses such data.

Against this backdrop, two major gaps exist in the literature. First, prior work using digital traces has studied clinical routines (Goh et al., 2011), invoice processing routines (Pentland et al., 2011), and garbage collection routines (Turner & Rindova, 2018). However, understanding how routines are shaped in production companies, i.e., production routines, has not received research attention. Compared to previously investigated routines, production routines have shorter turnaround times and must comply with strict regulatory standards. The latter aspect is especially intriguing as production routines typically have no flexibility in the sequence of performed actions and thus lend themselves to studying differences in the performance of routines when the patterns are fixed. Production workers are also often professionally and socially bonded as they may work side-by-side for 8 to 10-hour shifts. Additionally, production routines consume the majority of resources and are thus of critical importance for the success of manufacturing firms. Understanding how errors affect production routines would thus help streamline existing processes and improve production planning (Van Dyck et al., 2005).

Second, in investigating the mechanisms by which the performance of production routines can be affected, the role of errors as interruptive events has often been overlooked. Prior research notes that interruptions can shape organizational routines (Zellmer-Bruhn, 2003). Yet, the lack of objective and quantifiable data makes it often difficult to study the interplay between routines and errors, as perceptual data on errors is often plagued by biases, such as social desirability.

In fact, a recent paper (Carroll et al., 2021) highlights that the "analysis of 'big data' [...] can boost the generation of high-quality evidence concerning errors and their remedies." Frese & Keith (2015), emphesize how studying errors over time enhances error management strategies; however, the effect of errors on organizational routines has not been investigated. Therefore, scholars encourage empirical studies on errors to identify new error management strategies based on employee performance change vis-à-vis errors (Van Dyck et al., 2005). In the light of these research gaps, we address the following research question.

RQ: *Do errors affect production routines' performance, and if so, how?*

In this study, we leverage 75K complaints associated with 3M production batches over six years to examine the short- and long-term changes in production routines in response to errors. To do so, we use digital traces, a novel approach to studying business processes as is, i.e., how organizations actually run their processes —not how they are supposed to be executed (Van der Aalst, 2016). The processual knowledge gained from the interaction of human agents with IT, in the form of digital traces, gives an unprecedented opportunity to study key aspects of organizations, e.g., organizational routines, learning, and processes (Golder & Macy, 2014; Pentland, Recker, et al., 2020; Rahmandad et al., 2009).

More specifically, we use digital traces to obtain the production data at a large Dutch public holding company. We combine the data on production routines with process and order data to decompose routines into their structures and processes. The granularity of the data enables us to aggregate it at the routine level and focus on identical routines (pertaining to identical orders), where employees have limited flexibility to change the sequence of production steps. We create a daily panel data set of production routines tracking their performance and errors. We distinguish routine effectiveness and routine efficiency as the two main aspects of production routines' performance (Turner & Rindova, 2018). We use reporting *accuracy* (i.e., the degree to which employees fulfill the reporting requirements of each order) and *throughput time* (i.e., the time employees spend to process an order from the production start to the end in the production line) to measure routine effectiveness and routine efficiency, respectively.

This study makes two main contributions. First, it contributes to research on error management by studying the effects of errors (as an essential component of error management and culture) at a routine level using "big data." Specifically, it distinguishes between the effects of errors on production routines' performances with few vs. many prior errors and thereby highlights the nuanced implications of errors for employees' behavior. Second, it contributes to research on organizational routines by using digital traces to study production routine performance. Thus, it builds on and extends recent work on process multiplicity (Pentland, Recker, et al., 2020), i.e., the possible ways a process can unfold.

6.3 Theoretical Development

Numerous studies measure organizational performance in terms of effectiveness ("resource-getting ability") and efficiency ("amount of output obtained from a given input") (Davis & Peri, 2002, p. 87). Business processes play a pivotal role in reflecting the effectiveness and efficiency of organizations (Mans et al., 2013) because they guide the key operations performed by organizations and reflect their ability to adapt to changing environments.

To examine how errors affect the performance of production routines, we draw upon the literature on four areas: (1) organizational routines to study the collective performance of a group of people (i.e., routine participants) working together toward a business objective; (2) habit theory as regular and frequent interruptions; (3) error management to highlight how errors at the routine level can be leveraged as a source of learning and innovation; and (4) post-error behavior to disentangle the cognitive processes triggered by errors. The following sections present a brief review of these theoretical foundations.

6.3.1 Organizational Routines

Routines have been used to explain the stability and variability of organizations (Feldman & Pentland, 2003). Initial research on organizational routines introduced different images of routines, namely as a source of inertia and inflexibility (Aldrich, 1999; Baum & Singh, 1994; Winter & Nelson, 1982) versus as a source of flexibility and change (Adler et al., 1999; Feldman, 2000; Pentland & Rueter, 1994). More recent literature views organizational routines as dynamic and generative systems (Cohen et al., 1996; Feldman & Pentland, 2003; Hodgson, 2003; Lazaric, 2000; Lazaric et al., 2001; Pentland & Rueter, 1994). This perspective values the constantly changing nature of routines over their stability (Feldman, 2016). Organizational studies typically rely on routines to theorize aspects such as organizational capabilities and learning (Felin et al., 2012).

Whereas research on organizational routines focuses on organization-level performance, research on routine dynamics focuses on routines, per se, and the actions constituting them (Feldman et al., 2016). Routine dynamics takes a processual perspective to explain the stability and change of organizations (Feldman et al., 2016). According to this perspective, routines have two interdependent components, the ostensive (the enactment of actions, i.e., know-how) and the performative ("specific actions taken by specific people at specific times when engaged in a routine," i.e., know-what) components (Feldman & Pentland, 2003, p. 101). The dynamic outcomes of routines result from the interplay of these two aspects. Field studies show that action patterns are temporal (like a dynamic process with some extent of flexibility) and constantly subject to change, which is viewed as an inherent aspect of routines (Feldman et al., 2016). The dynamic nature of routines leads to various performances, i.e., executions do not necessarily become stable over time, and instead change in response to environmental circumstances.

Within the context of organizational routines, we borrow the term multiplicity to explain this study's focus. Pentland, Mahringer, et al. (2020, p. 5) define process multiplicity as "a duality of ‘one’ and ‘many’,” where “one” refers to one process and “many” refers to the space of possible paths to enact the process. Prior research notes multiplicity in routines’ performative and ostensive aspects. Whereas prior work often deals with multiplicity in the action patterns (ostensive) and their sequences (e.g., Turner & Rindova (2018)), we focus on multiplicity in only the performative aspect where patterns do not change and only the performance changes, for example, the process order is always the same, but the process throughput time varies (Mahringer & Pentland, 2021). Along similar lines, Pentland, Mahringer, et al. (2020) differentiate between performance and path (i.e., sequence of actions). Adopting their definition, Figure 4 contains four exemplary performances and two paths —one for performances 1 and 2 and the other for performances 3 and 4. This study focuses on the variation of multiple performances associated with a single path. Thus, we study routine participants who generate identical patterns by executing the same actions in the same order (i.e., the same path) with various performance measures (e.g., they sometimes perform the routine fast and sometimes perform it slowly). Generating identical patterns is common in capital intensive, process based, and heavily regulated industries. Anand et al. (2012) label such circumstances as “intended stability,” defined as the obligatory adherence to pre-planned operational routines that leads to consistent and conforming action patterns.

Like processes, routine performance measures vary in two crucial ways: routine efficiency and routine effectiveness (Turner & Rindova, 2018). Routine efficiency refers the utilization of resources, such as time and costs, in the execution of routines. Routine effectiveness refers to outcome quality, for example, whether it deviates from any regulations (e.g., order of actions, fulfillment of standard requirements) (Bapuji et al., 2019). We use these two terms because the context of this study does not allow for changes in action patterns. Thus, routine effectiveness is limited to complying with the quality of work rather than the order of process steps. We use this opportunity to focus on the multiplicity of the performative aspect, i.e., study the variation of routines’ performances when patterns must stay the same.

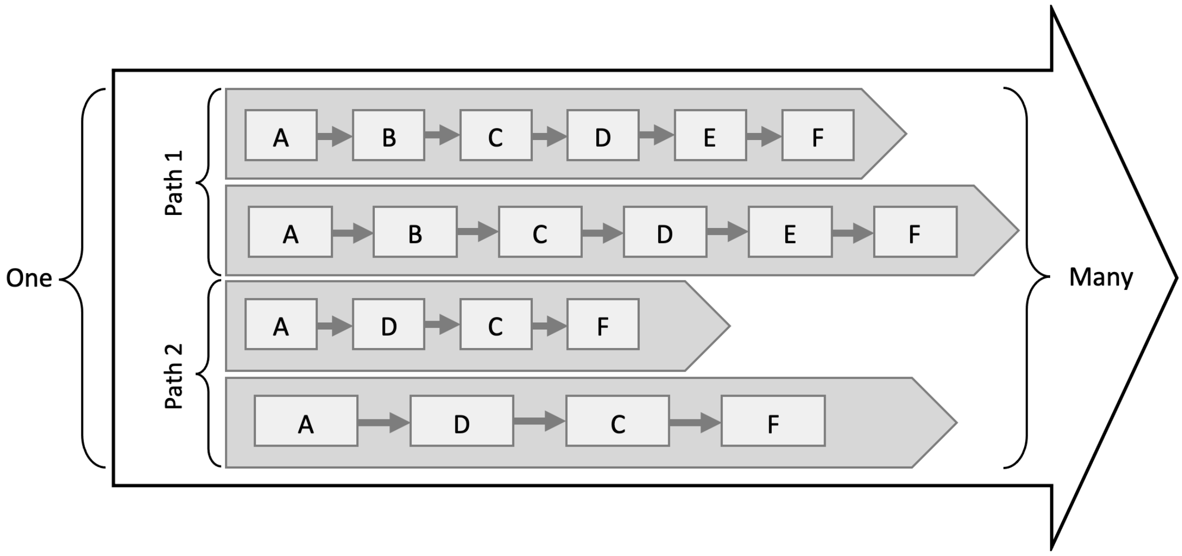


Figure 4: Process multiplicity for two paths and four performances; based on Pentland et al. (2020).

We rely on two measures from our digital traces that reflect routine effectiveness and efficiency. Reporting accuracy refers to the coverage ratio of fulfilling the reporting requirements by routine participants. Throughput time refers to the total time production employees spend processing a batch.[5]

6.3.2 Habits

Habits are commonly studied at the individual level, but a strong link exists between individuals' behavior and the routine in which they participate (Cohen, 2012). According to Wood & Neal (2009), repeated actions cued by the same context form habits. Applied to production routines, orders cue the actions of individuals in the context of their work environment and this context is not subject to regular changes due to the highly regulated nature of the manufacturing industry. Habits form when thoughtful and conscious actions gradually become automatic and unconscious. Thus, when employees undertake similar actions repeatedly, they adopt strong work habits. Accordingly, we draw from research on habits to analyze individuals' behavior (and change) in their respective production routines.

[5]Employees process one batch in each production round. They group small orders in a batch and break large orders into several batches. All pieces in a batch receive the same treatment.

Routine participants receive feedback on their performance shortly after the internal quality assurance team files a customer complaint. Errors play a critical role in employees' daily activities with consequences ranging from mandatory group training to the firing of certain employees. Moreover, habit-changing approaches usually introduce frequent and regular interruptions to effectively break habitual routines and form new habits (Zellmer-Bruhn, 2003). Interruptive events invoke a switch from automatic to conscious information processing (Louis & Sutton, 1991) We believe that frequent and regular errors play the commonly known interruption role at the routine level. However, this interruption occurs in the background as an unplanned action despite its significant impact on employee performance.

Looking at the big picture, routine participants develop individual habits by repeatedly conducting the same type of action. The collective behavior of routine participants can be the result of individuals' performance, or it can be different from individuals' performance due to their role in a team, i.e., individuals' reaction changes when they participate in teamwork. Individuals develop habits and thus switch their activities from conscious to unconscious while repeatedly working on the same activities. Errors interrupt this process due to its critical impact on employees' mindset.[6]

6.3.3 Error Management

Errors represent important components of organizational life and have been the subject of much prior organizational research (Carroll et al., 2021; Haunschild et al., 2015). The cost of errors can hurt organizations' reputations or lead to massive expenses. For example, in 2017, Honda recalled 800,000 minivans because of faulty seats; Samsung Securities Chief Executive Officer resigned after an employee made a $105 billion mistake in issuing stocks. Medical errors cause

[6]In the empirical analysis, we could not disentangle individual and team-level effects as we lack access to individual-level data. However, in several conversations with managers, we learned that individual frontline employees are key parts and determinants of routines. Alpha has dedicated professionals for each production step. So, the same individual is responsible to run the same activity (e.g., sealing) for every order the whole day. This ensures lower error rates because employees gradually become skilled professionals and make less mistakes.

over 250,000 deaths annually in the U.S. (Makary & Daniel, 2016). Air France 447 crashed due to human and machine errors (Oliver et al., 2017). Errors in organizations are defined as unintended and potentially avoidable deviations from predefined organizational plans that can negatively impact organizations (Carroll et al., 2021; Frese & Keith, 2015; Lei et al., 2016). Organizations' typical strategy toward errors is error prevention, i.e., eliminating errors before they happen and trying to predict errors to control their occurrence. In contrast, error management strategies suggest accepting errors as an inseparable part of organizations and, instead, controlling their negative consequences, e.g., by conducting training (Van Dyck et al., 2005). Given that errors cannot be fully predicted and controlled, organizations can integrate error management strategies to learn from errors and improve their systems and regulations (Carroll et al., 2021; Stern et al., 2008).

Many organizations implement error prevention strategies, but not all use additional error management practices. Researchers highlight that balancing error prevention and error management requires a deeper understanding of errors. The knowledge gained through post-error information gathering tends to be an important source for organizational learning (Desai & Madsen, 2022). In fact, research has shown that knowledge gained through errors depreciates more slowly than knowledge gained from successes (Madsen & Desai, 2010). Thus, organizations that implement only error prevention protocols (and no error management protocols) reduce their chances of learning from them (Sitkin, 1992). Errors in organizations can manifest themselves at the individual, team, or system level, but research has given greater attention to the individual level (Lei et al., 2016). Studying errors at any level helps organizations effectively balance their error prevention and error management activities. Scholars encourage more research on the dynamics of errors and response to errors over time (Carroll et al., 2021; Lei et al., 2016).

This study focuses on errors at the routine level, which is similar to the team-level in the organizational error literature, e.g., Bell & Kozlowski (2011). We study errors over time to understand how routines change in response to errors. Our view on errors mirrors that of error management scholars, i.e., errors cannot be eliminated but should be accepted as a manageable aspect of organizations.

6.3.4 Post-Error Behavior

As noted by cognitive science researchers, human behavior changes when their errors are detected (e.g., Damaso et al. (2020)). This literature considers two major post-error behavioral changes: (1) post-error slowing (PES) and (2) post-error improvement in accuracy (PIA) (Danielmeier & Ullsperger, 2011; Notebaert et al., 2009; Schroder & Moser, 2014). PES refers to individuals' tendency to spend more time on their current task after they made an error in their previous task. PES studies report the effect duration varying from several minutes to several months (e.g., Danielmeier & Ullsperger (2011), Segalowitz et al. (2010)). PIA refers to individuals' increased accuracy immediately after they made an error in their previous task.

This paper investigates whether the results of prior post-error studies apply at the team level. To our knowledge, prior literature lacks a comparable team-level analysis —possibly due to the complexity of production tasks compared to experimental research tasks. Despite these contextual differences, we base our initial hypotheses on the results of post-error studies. In the light of this background, we expect that routine participants respond to errors by slowing down their work pace to focus on accuracy.

6.3.5 Hypotheses

Habits are shaped when thoughtful and conscious actions gradually become automatic and unconscious (Wood, 2017; Wood & Neal, 2009). When errors are reported, we expect performance changes as individuals switch from automatic and unconscious actions to thoughtful and conscious ones. Employees prepare themselves for a serious discussion when being informed about an error or a complaint linked to their performance with potential measures ranging from group training to firing a team member. Therefore, we view errors as significant interruptions, which are likely to involve team-level performance changes (Zellmer-Bruhn, 2003). When individuals switch from automatic habits to informed habits (due to the interruption caused by the reported error), presumably, production slows down because conscious actions demand more mental effort (Wood & Neal, 2009). This assumption aligns with the post-error slowing effect discussed in the previous section (Notebaert et al., 2009). Thus, we expect individuals to slow down in response to errors but work with increased accuracy. In other words, individuals become more attentive to the work they perform, and, in turn, they require more time. So, we posit the following:

Hypothesis 1 (H1). *Previous-day errors are associated with an increase in routines' reporting accuracy.*

Hypothesis 2 (H2). *Previous-day errors are associated with an increase in routines' throughput time.*

As previously noted, errors play the interruption role similar to a habit-changing intervention. Prior studies on habit formation confirm that interruptions are effective if applied frequently and regularly (Graybiel et al., 2008; Wood, 2017; Wood & Neal, 2009). From a cognitive science perspective, Notebaert et al. (2009) link post-error performance changes to individuals' increased attention to infrequent events that direct attention away from the focal task. According to their findings, infrequent incidents automatically capture individuals' attention, regardless of the outcome being failure or success. In an experiment, they increase the number of errors compared to success and find the slowing effect only after successful attempts. They argue that individuals get distracted by the surprise of the incident.

Applied to the context of our study, some routines receive errors more frequently than others. Additional errors might become less distractive to employees for routines with more prior errors due to their repetitive nature. Accordingly, we anticipate different post-error consequences for routines with fewer prior errors than routines with more prior errors because frequent interruptions often lead to sustained changes. This effect is rooted in employees' exposure to frequent surprises in that they develop a tolerance against the distraction caused by an error. Thus, we argue that the previously mentioned post-error consequences are likely weaker for routines with more prior errors. The habits perspective also suggests that individuals would form new habits toward routines with more prior errors than routines with fewer errors. In other words, regular and frequent errors form a new habit that might neutralize immediate performance changes. So, we posit the following hypothesis, shown along with the other hypotheses in Figure 5.

Hypothesis 3 (H3). *The (a) increase in reporting accuracy and the (b) increase in throughput time after errors is greater among routines with fewer prior errors than routines with more prior errors.*

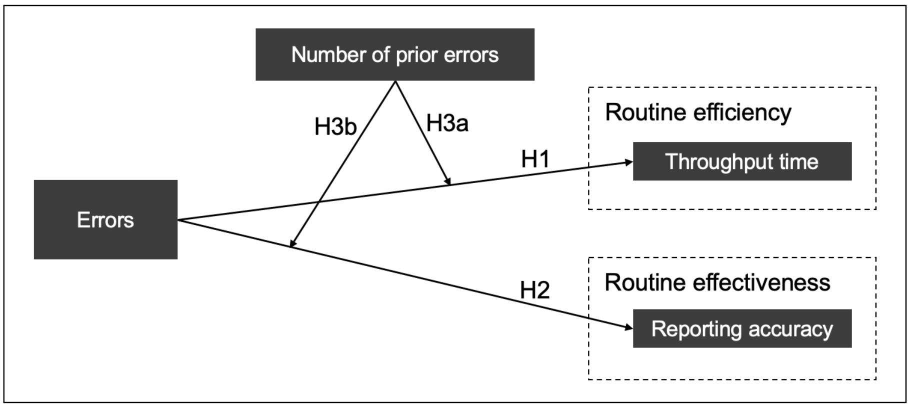


Figure 5: Research Hypotheses

6.4 Research Setting and Data

6.4.1 Study Context

We conducted this study within a large metal coating company, Alpha, which offers metal treatment services such as applying surface treatment services on unfinished products of customers from the automotive industry. Original equipment manufacturer (OEM) companies ship their unfinished products to Alpha to receive the final coating treatments. Alpha applies the required treatments and returns the finished products to the OEM. A large part of Alpha's revenue comes from long-term periodic orders that are placed regularly by the same customers. These periodic orders follow the same treatments to ensure consistency with previously delivered orders. Within Alpha's IT system, the detailed production specifications and steps of each order are referred to as article and each distinct article represents a routine in our study. An article might stay in production for several years.

Any deviation from the planned production process can significantly influence the final products. Coating companies therefore strictly control their production processes to ensure full compliance with regulatory demands. Thus, production steps must always follow the intended process order. Although regulations forbid changing the order of actions, no restrictions apply to the quality of performing the activities such as the time it takes to perform an action or the degree to which specific steps are recorded.

Alpha operates 15 plants in several European countries, including Germany, Poland, and the Netherlands. In these plants, Alpha uses a centralized IT system to ensure process compliance and monitor production performance. Alpha provides all frontline employees with a portable scanner and asks them to scan the production items twice, once when they start and once when they finish working on the respective items. These scanners generate granular data that we describe in the next section. Further contextual information on Alpha is given in Appendix C.1.

6.4.2 Data

We sourced the data through direct access to Alpha's internal IT systems. Our initial sample spans 3M batches and 371K unique articles. We refine the sample in five ways, as summarized in Table 16.

Table 16: Sample Selection Procedure

Step	Number of Batches	Number of Articles
Full sample (Jan 14 2013 to Mar 8 2021)	3,067,939	370,982
After restricting to years from 2016 to 2021	2,490,426	304,892
After restricting to articles with the first error between 2016 and 2021	943,243	33,861
After removing weekends and local holidays	900,988	33,654
After considering throughput times >= 10 minutes	796,675	32,648
After restricting to articles produced >= 10 times	717,916	7,818

First, we remove data before 2016 because many of the plants transitioned to new IT systems from 2013 to 2015. Thus, the data before 2016 may not be reliable because they were entered into the system post-hoc. Second, we only consider batches of articles that received their first error between 2016 and 2021. Articles associated with no errors or many errors before 2016 might differ in unobservable ways. Third, we remove all batches produced on weekends because fewer employees work on weekends, making the throughput times on weekends longer for batches produced compared to batches produced during the week. Moreover, Alpha usually stops most activities on weekends and only produces articles in rare exceptions, for example, if customers have urgent requests. Fourth, we discard all batches where throughput times are either missing or below ten minutes. Missing throughput times occur if one or both of the start and end timestamps are missing. Also, in interviews with production experts at Alpha, we learned that none of the batches at Alpha could be processed in fewer than ten minutes. This is an indication that throughput times below ten minutes do not accurately reflect actual throughput times. Fifth, of the remaining observations, we focus on articles that are produced at least ten times in our observation period to ensure that these articles are repeatedly produced and thus reflect the character of repeated actions.[7] We collapse the resulting 7,818 articles and 717,916 batches to a panel data set with 265,191 article-day observations. Each article is produced multiple times. Thus, our panel variable is the article identifier.

[7] In Table 27 in Appendix D, we show that our findings are qualitatively unchanged to filtering articles that are produced at least 20, 30, 40, and 50 times.

6.4.3 Dependent Variables

Each batch has a number of flagged actions that represent the key activities of an article and are mandatory to report. The dependent variable, reporting accuracy, represents the fraction of mandatory actions that were reported for article i on day t. Given that each article i can be produced in $1, 2, 3, ..., n$ batches b on day t, the reporting accuracy for article i on day t is calculated as:

$$ReportingAccuracy_{it} = \frac{\sum_{b=1}^{n} NumberofMandatoryActionsReported_{itb}}{\sum_{b=1}^{n} NumberofMandatoryActions_{itb}} \tag{22}$$

For example, if one batch of an article is produced and that batch requires four flagged actions to be reported, then frontline employees must report the four flagged activities to reach 100 percent. If they report three flagged steps, then reporting accuracy would be 75 percent and so on. We measure throughput time of article i on day t as the average number of hours between the start and end of the batches b initiated for article i on day t:

$$ThroughputTime_{it} = \frac{\sum_{b=1}^{n} [BatchProductionEnd_{itb} - BatchProductionStart_{itb}]}{n} \tag{23}$$

Thus, our measure of throughput time focuses on the core production process, i.e., applying the coating treatments, excluding the leading and trailing activities such as transportation, packaging, and shipment. Both dependent variables are undefined if article i was not produced on day t.

6.4.4 Independent Variable

We separately obtained data on 75K errors that were formally entered into Alpha's error management system. Each error is associated with one article. We match this data with the panel data to track the number of daily errors that were recorded for each article. In total, 28,417 errors were recorded for the articles in our sample. Table 17 presents the errors by causes, consequences, year, and plant.

Table 17: Description of Errors

Characteristics	Number of Errors	Characteristics	Number of Errors
Causes of Errors (Top 6)		Errors per Plant	
- Activation	1,251	- Plant01	572
- Clamping	2,104	- Plant02	1,767
- Flushing	2,829	- Plant03	879
- Handling	2,636	- Plant04	485
- Residuals	2,367	- Plant05	6,518
- Unknown	1,251	- Plant06	2,971
Consequences of Errors (if any)		- Plant07	8,300
- Corrective Actions	3,767	- Plant08	73
- Preventive Actions	2,417	- Plant09	1,509
Errors per Year		- Plant10	350
-2016	6,716	- Plant11	18
-2017	5,306	- Plant12	464
-2018	5,649	- Plant13	515
-2019	5,672	- Plant14	2,353
-2020	4,440	- Plant15	1,643
-2021	634		

Note:
This table reports descriptive statistics on the errors included in the sample. The numbers on causes and consequences of errors do not add up.

Our primary independent variable, any error, is measured by a dummy variable indicating whether any error has been reported for article i on day t. First error is a binary indicator that denotes one only for the first error reported for an article. Figure 6 plots the seven-day moving average of the sum of any error and first error. There is considerable variation over time in both variables. Figure 6a shows strong seasonal trends, with a notable peak for any error in September 2016 and a considerable drop at the height of the COVID-19 pandemic in the first half of 2020.[8] As expected, Figure 6b shows a downward trend for the sum of first errors. Thus, most articles receive their first errors toward the beginning of our observation period, which underscores that the articles in our sample are repeated and long-standing orders. If available,

[8]In Table 28 in Appendix D, we show that dropping either September 2016 or January to June 2020 did not influence our results.

we recorded the date on which an error has occurred, allowing us to separate the effects of error reporting and error occurrence. Moreover, we track whether errors are associated with corrective actions and/or preventive actions. Compared to conventional errors without actions, errors associated with actions hint at the identification of more deeply rooted shortcomings.

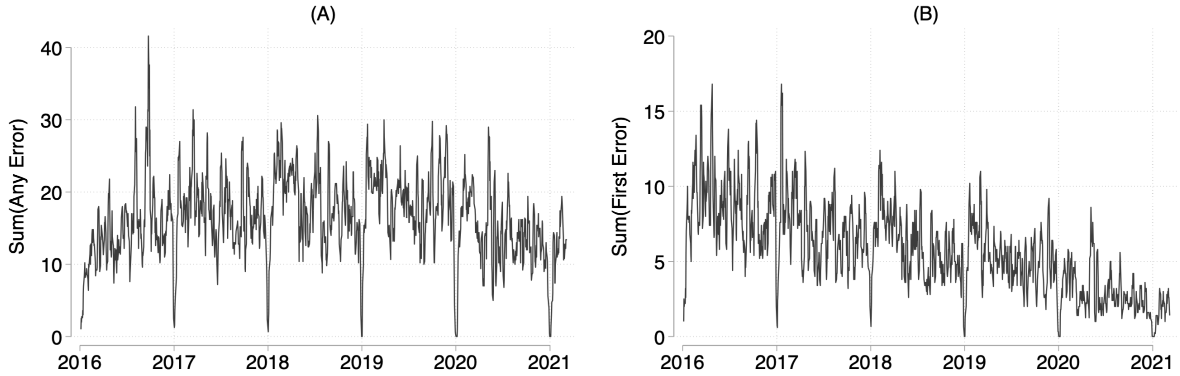


Figure 6: Any Error and First Error, 2016 to 2021

Note: The figure plots the seven-day moving average of the sum of any error and first error over the observation period from 4 January 2016 to 8 March 2021 (inclusive).

6.5 Results

6.5.1 Errors and Routine Performance

As a first test of the effect of errors on routine performance, we regress daily measures of throughput time and reporting accuracy for an article on lags and leads of errors reported for that article, controlling for seasonality, heterogeneity across plants, and previous performance. Our specification follows Durante & Zhuravskaya (2018) in that we include multiple lags and leads of our independent variable. In particular, we estimate equations of the following form:

$$R_{it} = \alpha_0 E_{it} + \beta_0 E_{i,t-1} + \sum_{\tau=1}^{7} \alpha_\tau E_{i,t+\tau} + \sum_{\tau=2}^{7} \beta_\tau E_{i,t-\tau} + \gamma_1 R_{i,\omega_t - 1} + P_i + \eta_{d_t} + \psi_{m_t} + \theta_{y_t} + \epsilon_{it} \quad (24)$$

The term R_{it} is a measure of routine performance (either reporting accuracy or throughput time) for article i on day t. E_{it} denotes the reporting of any error for article i on day t. We focus on the effect of same-day and previous-day errors with and without controls for a series of its lags and leads. The variable R_{i,ω_t-1} is a measure for the average routine performance for article i one week before (i.e., between 1 and 7 days before day t). P_i captures time-invariant plant fixed effects.[9] The terms η_{d_t}, ψ_{m_t}, and θ_{y_t} denote fixed effects for each day of the week, each calendar month, and each year, respectively.

Figure 7a plots the coefficients on the lags and leads of errors from estimating equation (24). It shows that the estimated effect of errors that occur the day before on reporting accuracy is positive and statistically significant. All the lags of errors taken together are jointly significant (Wald χ^2 = 3.76, p < 0.001), and all the leads are jointly insignificant (Wald χ^2= 1.02, p = 0.413), indicating that reporting accuracy only increases after an error is reported and that no spurious or erroneous associations exist before the error is reported. In fact, none of the seven leads are significant (at p = 0.05), which further corroborates the specific correlation between errors and reporting accuracy.

In Figure 7b, we show that the estimated effect of errors that occur the day before on throughput time is negative and statistically significant, which indicates that articles, on average, are produced faster after an error is reported. Again, all the lags of errors are jointly significant (Wald χ^2 = 5.07, p < 0.001). However, for throughput time the coefficients of several leads are positive and significant, and all leads taken together are jointly significant (Wald χ^2 = 11.68, p < 0.001).

Figure 10 presents plots based on regressions in which lags and leads of the errors are included in the regressions one by one instead of simultaneously. These patterns are identical to those in Figure 7. Thus, our results do not depend on errors and their lags (or leads) being included simultaneously.

[9] Our preferred specification uses plant fixed effects instead of article fixed effects because many articles have low within-cluster variation for reporting accuracy. In Tables 29 and 30 in Appendix D, we present the results of our regressions with article fixed effects. The results are largely consistent.

Table 18 presents the results formally. The effect of errors on reporting accuracy and throughput time are presented in columns 1-3 and columns 4-6, respectively. Columns 1 and 4 present the relationship between routine performance and errors conditional only on day-of-the-week, calendar-month, and year fixed effects. The results indicate that neither reporting accuracy nor throughput time are significantly correlated with errors on the same day. But if we include previous-day errors in the list of covariates; the coefficient on yesterday's error on today's reporting accuracy is positive and statistically significant. By contrast, we find no association between previous-day errors and today's throughput time. We present these results in columns 2 and 5, respectively. In columns 3 and 6, we add controls for previous routine performance. The effect of errors at $t-1$ on reporting accuracy on day t remains positive and statistically significant. In addition, the effect of errors at $t-1$ on throughput time becomes negative and significant.[10]

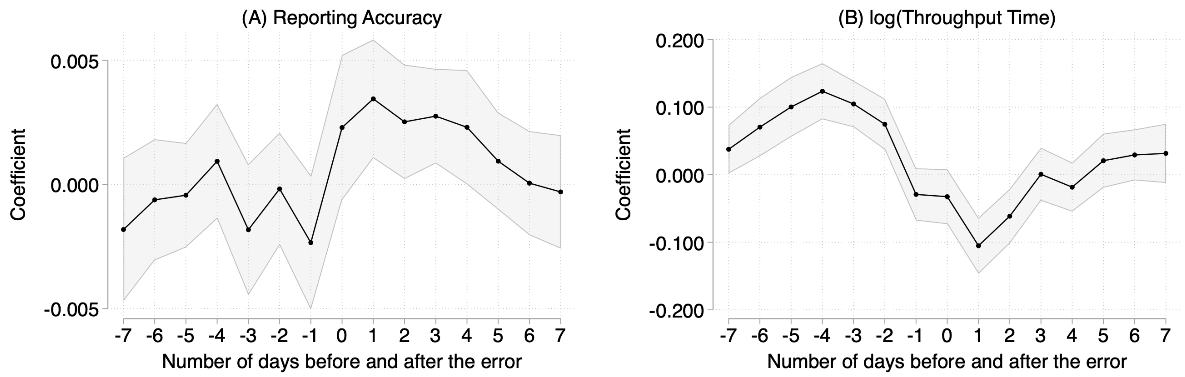


Figure 7: Any Error and Routine Performance Including Seven Leads and Lags

Note: The figure reports the estimated coefficients (and respective 95 percent confidence intervals with standard errors clustered by article) from the regressions of reporting accuracy and throughput time on any reported errors between 7 days before and 7 days after the event. The left panel uses reporting accuracy as the dependent variable and the right panel uses the log-transformed throughput time in hours. The list of covariates also includes year, month, and day-of-the-week fixed effects, and controls for the average reporting accuracy (fig. A) and throughput times (fig. B) 1 week before. Results come from cols. 1 and 3 of Table E.1.

[10]To account for the possibility of information leakage before error reporting, we repeat all the analyses with leading (i.e., $t+1$) value of error reporting. None of the results change with this inclusion. The detailed results are excluded due to space considerations but available from the authors upon request.

Table 18: Any Error and Routine Performance

Dependent Variables	Reporting accuracy			log(Throughput time)		
Model	(1) OLS	(2) OLS	(3) OLS	(4) OLS	(5) OLS	(6) OLS
Any $Error_t$	0.001	0.001	0.002+	0.012	0.009	0.003
	(0.001)	(0.001)	(0.001)	(0.039)	(0.036)	(0.022)
Any $Error_{t-1}$		0.005***	0.004**		0.049	-0.066**
		(0.001)	(0.001)		(0.046)	(0.024)
Reporting $Accuracy_{[Prev.-WeekAvg.]}$			0.786***			
			(0.018)			
log(Throughput $Time)_{[Prev.-WeekAvg.]}$						0.691***
						(0.012)
Time Fixed Effects	Yes	Yes	Yes	Yes	Yes	Yes
Plant Fixed Effects	Yes	Yes	Yes	Yes	Yes	Yes
Number of Observations	265,191	265,098	141,500	265,191	265,098	141,500
Number of Articles	7,818	7,818	5,965	7,818	7,818	5,965
Adj. R-squared	0.247	0.247	0.668	0.196	0.196	0.523

Note:
OLS regressions are presented in cols. 1-6. All regressions include year, month, and day-of-the-week fixed effects. Standard errors clustered by article are reported in parentheses. The number of observations differs because of the inclusion of lags and previous week averages in cols. 2, 3, 5, and 6. *** $p < 0.001$, ** $p < 0.01$, + $p < 0.10$.

6.5.2 Robustness

We conduct the following robustness checks. First, Table 23 reports the results for an extended window of 14 leads and lags, which are consistent with the main results. Second, we test whether the results are robust to alternative operationalizations of errors. Table 24 reports the results for the log-transformed number of errors per day instead of a binary measure. Our findings are robust to this alternative measure. Third, we test whether our results are robust to alternative operationalizations of reporting accuracy and alternative estimators. We transform reporting accuracy into a binary variable that denotes 1 if all steps are reported and denotes 0 if any of the steps have not been reported. Column 1 of Table 25 show the results of the linear probability model using plant and time fixed effects. In column 3, we additionally control for previous performance of reporting accuracy. The results are consistent. Columns 2 and 4 of Table 25 show the results of the regressions using a Logit estimator. We find the results for the model using only plant and time fixed effects confirmed (col. 2). When controlling for previous reporting accuracy performance, the association between previous-day errors and the binary measure of reporting accuracy is positive but not significant (col. 4). Taken together, these analyses — although with one exception — support the main premises of our research. Fourth, we test whether our results are robust to additional control variables. In columns 1 and 4 of Table 26, we include a linear time trend in place of time fixed effects, in columns 2 and 5, we add the log-transformed number of cases as an additional control variable.

Finally, in columns 3 and 6, we control for the average performance of the previous 2 weeks (instead of 1 week). Our findings are robust to including these additional control variables.

6.5.3 Error Occurrence

So far, we have investigated the association between errors and behavior around the time when errors were formally entered into the error reporting system. A central premise of our analysis is that employees change their behavior when a complaint is filed and members of the quality assurance team make employees aware of their shortcomings. If this is indeed the mechanism by which the observed behavioral responses are elicited, we would expect no change in performance after an error has occurred. The idea is that individuals either (a) are not aware at the time that an error has occurred or (b) have no incentive to change their behavior after they make a mistake because there is a degree of uncertainty whether it will be noticed. In either case, no significant correlation between error occurrence and routine performance would support the assertion that people change their behavior only if errors are formally recorded by the organization and employees are held accountable for their mistakes.

To tease out the effect of error occurrence versus error reporting, we exploit the fact that we know for 4,088 errors when they occurred. On average, errors are reported 18.34 days (standard deviation = 43.44) after they have occurred. The results are presented in columns 1 and 4 of Table 19. We show that neither error occurrence nor its lag are correlated with reporting accuracy (col. 1) and throughput time (col. 4). These results provide strong evidence that the changes in routine performance are more likely to occur after an error is reported and not when it occurs.

Table 19: Error Occurrence, Prevention, and Correction and Routine Performance

Dependent Variables	Reporting accuracy			log(Throughput time)		
Model	(1) OLS	(2) OLS	(3) OLS	(4) OLS	(5) OLS	(6) OLS
Error Occurred$_t$	-0.006			0.045		
	(0.007)			(0.078)		
Error Occurred$_{t-1}$	0.010			0.096		
	(0.008)			(0.071)		
Error Corrective Action$_t$		0.008***			-0.010	
		(0.002)			(0.030)	
Error Corrective Action$_{t-1}$		-0.001			-0.170***	
		(0.002)			(0.030)	
Error Preventive Action$_t$			0.013***			0.043
			(0.004)			(0.056)
Error Preventive Action$_{t-1}$			-0.000			-0.090+
			(0.005)			(0.053)
Reporting Accuracy$_{[Prev.-WeekAvg.]}$	0.786***	0.786***	0.786***			
	(0.018)	(0.018)	(0.018)			
log(Throughput Time)$_{[Prev.-WeekAvg.]}$				0.691***	0.691***	0.691***
				(0.012)	(0.012)	(0.012)
Time Fixed Effects	Yes	Yes	Yes	Yes	Yes	Yes
Plant Fixed Effects	Yes	Yes	Yes	Yes	Yes	Yes
Number of Observations	141,500	141,500	141,500	141,500	141,500	141,500
Number of Articles	5,965	5,965	5,965	5,965	5,965	5,965
Adj. R-squared	0.668	0.668	0.668	0.523	0.523	0.523

Note:
OLS regressions are presented in cols. 1-6. All regressions include year, month, and day-of-the-week fixed effects. Standard errors clustered by article are reported in parentheses. *** $p < 0.001$, + $p < 0.10$.

6.5.4 Corrective Actions and Preventive Actions

The organization has implemented ISO 9001 as a quality management system. In line with ISO 9001, a subset of errors is associated with corrective actions and preventive actions. According to ISO (*ISO/DIS 9001*, 2014, p. 6), corrective action is to "eliminate the case of a nonconformity and to prevent recurrence" and preventive action is taken to "prevent the occurrence." In our sample, corrective actions are typically associated with rework and preventive actions often involve mandatory trainings. Errors which require corrective or preventive actions are typically more severe as they reflect the necessity for changes in the way that certain activities are carried out. Therefore, we expect such errors to trigger a more direct link of communication between quality management and front-line employees who carry out the work.

The results are shown in columns 2, 3, 5, and 6 of Table 19. In line with this explanation, we find that improvements in reporting accuracy materialize immediately on the day errors with a corrective action (col. 2) and/or preventive action (col. 3) are reported whereas the lagged variable is no longer significant. This indicates that errors that require changes to current procedures are most effective in improving compliance on the day that the news are spilled. Interestingly, changes in throughput time still occur one day after the error both for errors with corrective actions (col. 5) and preventive actions (col. 6).

6.5.5 Prior Errors

Next, we test the effect of prior errors on throughput time and reporting accuracy. The results are shown in Table 20. In columns 1 and 4, we replicate our main results and show that the lag of the first error is positively related to reporting accuracy ($p<0.001$) and negatively related to throughput time ($p<0.001$).[11] The magnitudes of the coefficients of first error are substantially larger than for any error, which presents a first test of our hypothesis that errors have a more pronounced impact on routine performance if routines have experienced fewer prior errors. In columns 2 and 5, we estimate the effect in a regression framework using the following model:

$$R_{it} = \beta_0 E_{i,t-1} + \beta_1 L_{i,t-1} + \beta_2 E_{i,t-1} \times L_{i,t-1} + \gamma_1 R_{i,\omega_t-1} + P_i + \eta_{d_t} + \psi_{m_t} + \theta_{y_t} + \epsilon_{it} \quad (25)$$

The term $L_{i,t-1}$ is a dummy variable that is set to 1 if the number of prior errors for article i was low, i.e., no more than one prior error (equal to the median of our sample) up to day $t-1$, and 0 otherwise. This helps us test whether employees react differently to articles with only few errors compared to many errors prior to the focal error. Columns 2 and 5 of Table 20 show

[11] In columns 5 and 6 of Table 25, we show the robustness of the relationship between first error and reporting accuracy using the binary operationalization of the dependent variable.

Table 20: Prior Errors and Routine Performance

Dependent Variables	Reporting accuracy			log(Throughput time)		
Model	(1) OLS	(2) OLS	(3) OLS	(4) OLS	(5) OLS	(6) OLS
First Error$_t$	0.007*			-0.033		
	(0.004)			(0.042)		
First Error$_{t-1}$	0.008***			-0.194***		
	(0.002)			(0.043)		
Any Error$_{t-1}$		0.003*	0.007**		-0.075**	-0.195***
		(0.001)	(0.002)		(0.028)	(0.039)
Low$_{t-1}$		-0.000			-0.070***	
		(0.001)			(0.016)	
Any Error$_{t-1}$ $\times$Low$_{t-1}$		0.005*			-0.057	
		(0.003)			(0.054)	
log(1+Prior Errors)$_{t-1}$			-0.000			0.040***
			(0.000)			(0.009)
Any Error$_{t-1}$ $\times$log(1+Prior Errors)$_{t-1}$			-0.001+			0.036*
			(0.001)			(0.016)
Reporting Accuracy$_{[Prev.-WeekAvg.]}$	0.786***	0.786***	0.786***			
	(0.018)	(0.018)	(0.018)			
log(Throughput Time)$_{[Prev.-WeekAvg.]}$				0.691***	0.689***	0.690***
				(0.012)	(0.012)	(0.012)
Time Fixed Effects	Yes	Yes	Yes	Yes	Yes	Yes
Plant Fixed Effects	Yes	Yes	Yes	Yes	Yes	Yes
Number of Observations	141,500	141,500	141,500	141,500	141,500	141,500
Number of Articles	5,965	5,965	5,965	5,965	5,965	5,965
(Adj.) R-squared	0.668	0.668	0.668	0.523	0.524	0.524
Test of the coefficient difference of						
First Error$_{t-1}$ and Any Error$_{t-1}$						
SUEST chi^2-value	4.82			10.33		
SUEST p-value	0.0282			0.0013		

Note:
OLS regressions are presented in cols. 1-6. All regressions include year, month, and day-of-the-week fixed effects. Standard errors clustered by article are reported in parentheses. The last two rows present the chi2-values and p-values for equality between the effects of First Error(t-1) and Any Error(t-1) (presented in cols. 3 and 6 of Table 12) calculated using seemingly unrelated estimation tests (SUEST). *** p $<$ 0.001, ** p $<$ 0.01, * p $<$ 0.05, + p $<$ 0.10.

significant interactions between any error and low, which indicates that the association between errors and routine performance is stronger for articles associated with only few prior errors.

In columns 3 and 6, we show the robustness of our results using the log-transformed number of prior errors instead of a dummy variable. We find consistent results in the sense that the coefficients on the interaction effects have the opposite signs, suggesting that the relationship between errors and routine performance becomes weaker as the number of prior errors increases.

6.5.6 Endogeneity Tests

In the previous subsection, we established a robust association between errors and routine performance. However, our results could be biased because of endogeneity: both errors and routine performance may be driven by a third unobserved variable. For example, issues with a production line may have affected errors and they directly affected routine performance related to that article. Thus, it is instructive to use an identification strategy that is more robust to unobserved variables.

We exploit the fact that customers and articles have one-to-many relationships, that is each customer can order different articles but each article is uniquely associated with one customer. We implement an instrumental variable (IV) specification similar to that used by (Bartik, 1991) and (Card, 2001). Specifically, we instrument for errors reported for article i on day $t-1$ using the following measure:

$$IV_{i,t-1} = \frac{E_{pj,2016}}{E_{j,2016}} \times \left[E_{j,t-1} - E_{i,t-1}\right] \tag{26}$$

This IV measure has two components. The term in parentheses represents the errors reported by customer j on day $t{-}1$, excluding article i's contribution. This purges the measure of customer j's errors from article-specific factors. Instead, the fluctuations in errors reported by customer j for other articles — which are assumed to be driven by factors exogenous to i — drive the IV shocks to errors reported for article i.

In Table 31, we show a sample of three errors reported by a customer on 29 March 2017. Each error has a different error number, description, and article. This shows that a customer may report on multiple issues on the same day, indicating that the timing of other errors reported by that customer is likely to be exogeneous to the conditions related to the focal article i. Instead, there may be specific days when customers get in touch with the company to discuss issues related to their orders.

We also assume that some articles are likely to receive a larger share of errors reported by customer j. To weight the term in brackets, we use the plant p's share of the total customer errors in a base period. As each article is produced at one plant, articles produced at plants with many customer errors in the base period are apt to experience a higher share of the errors reported by customer j.[12] We use 2016, the first year of our data, as the base. We estimate the following equation with 2SLS:

$$R_{it} = \beta_0^{IV} E_{i,t-1} + P_i + \eta_{d_t} + \psi_{m_t} + \theta_{y_t} + \epsilon_{it} \tag{27}$$

where the previous-day errors E_{t-1} is instrumented by the first lag of our IV measure. We use a more parsimonious specification in this analysis because we have no independent instrument for same-day and next-day errors. Table 21 presents the results. Column 1 present the first-stage relationships for previous-day errors. The instrument is a very strong predictor of any error for article i at $t-1$. Columns 2 and 4 report the results of the second stage for reporting accuracy and throughput time using OLS regression with the reduced sample. The association between errors and reporting accuracy remains positive and marginally significant, whereas we do not find a significant association between errors and throughput time. Columns 3 and 5 report the results using 2SLS. The Cragg-Donald Wald F-test statistic rejects the null hypothesis of a weak instrument (Cragg & Donald, 1993). The value is 3485.52 and well above the threshold of 16.38 proposed by Stock and Yogo (2005). The Kleibergen-Paap rk Wald F-statistic is 37.43, which also mitigates concerns over a weak instrument (Kleibergen & Paap, 2006). The results are consistent with Table 18 and show a significant increase in reporting accuracy and a marginally significant decrease in throughput time.

[12] The results are unchanged when adjusting the share of errors in the base period for the production volume, V: $(\frac{E_{pj,2016}}{V_{pj,2016}})/(\frac{E_{j,2016}}{V_{j,2016}})$.

Table 21: IV Regressions and Previous-Day Errors

	Error_{t+1}	Reporting accuracy		log(Throughput time)	
Model	(1) OLS	(2) OLS	(3) 2SLS	(4) OLS	(5) 2SLS
IV_{t-1}	0.020***				
	(0.003)				
Any Error_{t-1}		0.003+	0.028***	-0.047	-0.180+
		(0.002)	(0.008)	(0.032)	(0.108)
$\text{Reporting Accuracy}_{[Prev.-WeekAvg.]}$		0.746***	0.746***		
		(0.025)	(0.025)		
$\text{log(Throughput Time)}_{[Prev.-WeekAvg.]}$				0.698***	0.698***
				(0.016)	(0.016)
Time Fixed Effects	Yes	Yes	Yes	Yes	Yes
Plant Fixed Effects	Yes	Yes	Yes	Yes	Yes
Number of Observations	91,555	91,555	91,555	91,555	91,555
Number of Articles	4,090	4,090	4,090	4,090	4,090
(Adj.) R-squared	0.05	0.603	0.601	0.548	0.548

Note:
The table presents IV regressions with weighted number of errors for other articles by the same customer used as an instrument for the reporting of any error. The sample used in col. 1 is derived from the sample in cols. 2-5. All regressions include year, month, and day-of-the-week fixed effects. Standard errors clustered by article are reported in parentheses. *** $p < 0.001$, + $p < 0.10$.

In Appendix F, we show that our results are robust to using a difference-in-differences (DID) approach. We use one-on-one nearest neighbor matching to reduce heterogeneity between treatment and control groups. Taken together, our results suggest that endogeneity should not be a serious concern.

6.6 Discussion and Conclusion

This paper has focused on understanding the effect of errors on frontline employees' performance. Using a novel data set of production routines, we find robust evidence that production teams tend to work more accurately and faster after an error has been reported. The observed effect, however, disappears when errors become frequent. We split the performance of routine participants into effectiveness and efficiency as we theorize that employees might react to errors by working more effectively but less efficiently, i.e., they become more attentive to regulations and hence work slower. Surprisingly however, our analysis indicates performance increases for both effectiveness (higher reporting accuracy) and efficiency (shorter throughput time). These results suggest that frontline employees perceive errors as critical incidents and adapt their behavior accordingly, underscoring the positive implications of such interruptive events (Zellmer-Bruhn, 2003). However, employees only take errors seriously initially, but after receiving them frequently, they tend to form new habits in which errors become an integral aspect. Further research is needed to examine the interplay between errors and employees' performance, such as by considering the influence of additional factors like feedback delay (Rahmandad et al., 2009) and individual status (Koster & Aven, 2018).

Furthermore, we isolated the performative aspect of routines from the ostensive aspect in a large organization practicing the strict process execution of orders. Specifically, we used articles as specific and predefined production recipes to track orders with the same production path. Despite following the same process, long-term, multipart, and periodic orders varied in terms of their performance. This unique setting allowed us to elaborate on process multiplicity of inflexible production processes. Thus, we contribute to the literature on organizational routines by reporting new insights on the dynamics of performative patterns pertained to production routines. Our study responds to Pentland and Feldman's (2005) call to conduct empirical studies on variations of performance effectiveness and efficiency at the routine level. The results confirm the dynamic nature of organizational routines and shed greater light on the action patterns by reporting performance changes in response to errors.

We contribute to the error management literature by studying how the effects of errors unfold over time. Modern error management systems (e.g., complaint management systems) provide unprecedented access to granular data on errors. We use this opportunity to examine team-level performance changes in response to errors. Prior works (e.g., Carroll et al. (2021), Van Dyck et al. (2005), Frese & Keith (2015)) anticipate these results and call for further research on this topic. Our contribution to the error management literature is twofold. First, we confirm the constructive potential of errors in empirical research (Desai & Madsen, 2022; Madsen & Desai, 2010). We report on two essential components of routine performance separately, i.e., efficiency and effectiveness. Second, we elaborate on the circumstances associated with the main effect by tracking routine performance over time. We observe that the post-error effects fade out when errors occur frequently. These results suggest the necessity of effective error management policies to avoid the negative impacts of errors and maximize their potential for organizational learning. Future research may consider how errors shape performance in the presence of employee turnover (Joseph et al., 2022) or intentional employee misconduct (Larkin et al., 2021).

This study has several limitations. First, it is limited to the coating industry. We extended our data set to cover data from multiple plants in various European locations to ensure generalizability in similar contexts. However, as production routines are often context-specific and strictly regulated, our results may not apply to other types of routines, such as procurement or clinical routines. Our analysis also does not account for specific contextual factors. To our knowledge, the company did not face any significant managerial or structural changes during our study period. Nonetheless, we cannot conclusively rule out other possibly relevant factors, such as changes in plant managers or increased employee turnover.

Second, we base our analysis on digital traces collected from the direct interaction of humans with IT systems. Several hardware restrictions (e.g., plant design, sensors), as well as software controls (e.g., enforcing action orders, and accepting only predefined data ranges), might filter out possible outliers or erroneous cases. Nevertheless, the available data are still subject to error, noise, and common data quality issues. To mitigate these issues, we applied thorough data cleansing operations to minimize possible biases. We also performed several robustness and endogeneity tests to rule out potential confounds.

Third, scholars on organizational routines label them as complex and dynamic, and requiring more empirical research (Feldman et al., 2016). We respond to this call by measuring the performance of production routines in terms of their effectiveness and efficiency. We do not claim that our dependent variables, reporting accuracy and throughput time, are the only way to measure routine performance. Instead, we use these two variables due their reliability in terms of data quantity and quality. Thus, future research should extend our analysis to incorporate additional variables, such as customer ratings.

In conclusion, this study offers implications for managers involved in production routines. They should not underestimate the effect of errors on employees' performance. Our results suggest viewing errors as a new source of innovation, change, and productivity gains if leveraged effectively. When teams learn about their first errors, they become more efficient and effective, but this potential seems to fade out in the long term if errors become frequent. Thus, managers must strike a balance between error prevention and error management. Occasional errors seem to improve performance but recurring mistakes are less likely to "bother" employees in addition to being detrimental to customer satisfaction. So, managers should seek to prevent repetition of errors by acting quickly and effectively when they start to reoccur.

7 Limitations and Future Research

This dissertation delves into the challenges associated with analyzing event log data collected from enterprise systems and presents innovative solutions to address them. The primary focus is on process event logs, and each study approaches the data from a distinct perspective, tackling specific data analysis challenges. By blending event log control flow data with contextual information such as production and error reporting data, the results yield more tangible and business-relevant insights.

In the first paper, the application of trace clustering at the data preprocessing stage is explored, offering detailed insights into algorithm adjustments to handle potential complexity issues effectively. The second paper introduces a novel inefficiency index, capturing various dimensions of process inefficiency not addressed by previous tools and studies. This index enables enterprises to identify root causes and implement effective control measures. The fourth paper enriches the analysis with contextual data and draws upon theoretical insights from organizational science and error management studies. The findings shed valuable light on the impact of error communications on employee performance, utilizing event log data extracted from production processes.

Collectively, all studies within this dissertation contribute to the overarching research question (ORQ) by proposing novel methods, artifacts, experimental settings, and frameworks to extract new insights from event log data. The contributions encompass reducing complexity in event logs (paper one), measuring process inefficiencies (paper two), capturing event log measures (paper three), and exploring the effect of error communications on employee performance (paper four). The dissertation offers both generic and specific solutions to the identified problems, while thoroughly validating the results. It emphasizes that the prospects for research in this domain are limitless and acknowledges that this work does not claim to cover all potential issues within the vast research venue. Instead, it lays the foundation for future explorations and advancements in leveraging event log data to enhance organizational understanding and performance.

This dissertation acknowledges several limitations. One primary limitation pertains to the lack of research on the integration of contextual data attributes in the research on digital traces. While paper one explores the utilization of contextual attributes and domain expertise to enhance trace clustering results, the existing literature lacks practical and theoretical works that extensively elaborate on integrating contextual information into their approaches. Future research must prioritize addressing this gap to enhance the applicability and robustness of trace clustering methodologies.

Furthermore, the inefficiency index proposed in paper two solely considers the control-flow behavior of process traces, neglecting the intertwined nature of inefficiency with business context, such as process throughput time. Integrating contextual attributes into the inefficiency index could potentially yield more comprehensive findings. Paper two identifies this as a future research avenue, highlighting the significance of incorporating contextual elements to improve the accuracy and insights of inefficiency measurements.

In paper four, the study effectively utilizes two contextual attributes, throughput time and reporting accuracy, in addition to digital trace data. The results underscore the critical role of contextual attributes in capturing the true effects and implications of process deviations. Integrating further context into the analysis could offer enhanced explainability and a more comprehensive understanding of digital trace patterns.

One common limitation pertains to the absence of techniques for incorporating contextual attributes of event logs. This limitation potentially restricts the ability to comprehensively account for the contextual nuances that could impact the interpretation and outcomes of the analyses (Bose et al., 2013). Furthermore, another limitation concerns the accuracy of the event log data itself, which is inherently tied to the quality of the digital traces collected (Suriadi et al., 2017). As a specific instance, the reliability of timestamps in reflecting the actual duration of work remains circumscribed, thereby potentially introducing an element of distortion into the analyses. These limitations collectively underscore the need for enhanced approaches to incorporate contextual variables and enhance the precision of event log data, both of which would contribute to refining the validity and robustness of the findings across the examined research papers.

Another limitation lies in the availability of real data sets for testing the generalizability of novel approaches to studying digital traces proposed in numerous studies. While these studies showcase promising methodologies, the lack of diverse real-world data sets hinders the ability to validate their effectiveness in different contexts. Additionally, concerns regarding data protection and the value of academic-industry collaboration raised by industry partners further complicate the acquisition of real data sets and hinder qualitative data collection, which could complement quantitative studies.

In light of these limitations, conducting supplementary qualitative research emerges as a future research opportunity to enhance the findings of this dissertation. Qualitative research can provide valuable insights into complex phenomena, such as routine dynamics and process inefficiencies, offering a more comprehensive understanding and triangulating the quantitative findings derived solely from digital trace data.

Overall, this dissertation acknowledges the identified limitations and emphasizes the importance of addressing them in future research endeavors to strengthen the validity, applicability, and insights drawn from the analysis of digital traces in various domains.

8 Discussion and Conclusion

In conclusion, this dissertation delves into the domain of processual data analytics, focusing on the analysis of event logs from enterprise systems to extract valuable insights. It makes significant contributions to the field by addressing selected research topics and offering innovative methods and solutions relevant to both business researchers and practitioners.

The first paper provides a comprehensive review of process trace clustering literature and presents a generic framework, enabling researchers and practitioners to enhance their trace clustering approaches effectively. By breaking down complex process event logs into more manageable sub-event logs, trace clustering reduces complexity and enhances business analytics outcomes.

The second paper introduces a novel inefficiency index, capturing various dimensions of process inefficiency beyond commonplace industrial tools. It leverages trace features, extracted from digital traces, to measure inefficiency in the control flow of processes. Additionally, the third paper introduces a dedicated artifact, Fig4PM, serving as a library of 73 control-flow features extracted from the process mining literature, streamlining the implementation and reusability of trace features.

The fourth paper explores the effect of errors on frontline employees' routine performance. This study identifies routines with identical execution paths and compares their performative aspects, focusing on routine effectiveness and efficiency. The results suggest that employee performance increases in terms of accuracy and speed when informed about errors, although this effect diminishes with excessive error communications. These findings hold significant implications for employee communication practices, particularly error management policies.

Overall, the dissertation contributes to the field of process mining while differentiating between action patterns and performance patterns within trace data analysis. This differentiation sheds light on new avenues for future research, particularly in the exploration of organizational routines and routine dynamics. Scholars can leverage these findings to further develop the topic of process multiplicity and facilitate research on processual data in organizational settings.

As a whole, this dissertation emphasizes the potential of analyzing event logs to gain comprehensive insights into business processes, paving the way for data-driven decision-making and organizational improvement. The contributions made in this research set the stage for further advancements in processual data analytics, ultimately driving greater process efficiency, effectiveness, and understanding of organizational matters.

References

Accorsi, R., & Stocker, T. (2011). Discovering workflow changes with time-based trace clustering. *International Symposium on Data-Driven Process Discovery and Analysis*, 154–168.

Adler, P. S., Goldoftas, B., & Levine, D. I. (1999). Flexibility versus efficiency? A case study of model changeovers in the toyota production system. *Organization Science*, *10*(1), 43–68.

Aghabaghery, R., Hashemi Golpayegani, A., & Esmaeili, L. (2020). A new method for organizational process model discovery through the analysis of workflows and data exchange networks. *Social Network Analysis and Mining*, *10*(1).

Aldrich, H. (1999). *Organizations evolving*. Sage.

Anand, G., Gray, J., & Siemsen, E. (2012). Decay, Shock, and Renewal: Operational Routines and Process Entropy in the Pharmaceutical Industry. *Organization Science*, *23*(6), 1700–1716.

Appice, A., & Malerba, D. (2015). A co-training strategy for multiple view clustering in process mining. *Ieeexplore.ieee.org*.

Appice, A., & Malerba, D. (2016). A co-training strategy for multiple view clustering in process mining. *Transactions on Services Computing*, *9*(6), 832–845.

Ariouat, H., Cairns, A. H., Barkaoui, K., Akoka, J., & Khelifa, N. (2016). A two-step clustering approach for improving educational process model discovery. *2016 Ieee 25th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (Wetice)*, 38–43.

Augusto, A., Mendling, J., Vidgof, M., & Wurm, B. (2022). The connection between process complexity of event sequences and models discovered by process mining. *Information Sciences*, *598*, 196–215.

Aull, A. (2020). *3 leading indicators of on-time delivery you shouldn't ignore*. Celonis Blog. https://www.celonis.com/blog/3-leading-indicators-of-otd

Back, C. O., Debois, S., & Slaats, T. (2019). Entropy as a measure of log variability. *Journal on Data Semantics*, *8*(2), 129–156.

Back, C., Debois, S., & Slaats, T. (2018). Towards an entropy-based analysis of log variability. *Business Process Management Workshop*, *308*, 53–70.

Badakhshan, P., Gosling, S., Geyer-Klingeberg, J., Nakladal, J., Schukat, J., & Gsenger, J. (2020). Process mining in the coatings and paints industry: The purchase order handling process. *International Conference on Process Mining (ICPM)*.

Badakhshan, P., Wurm, B., Grisold, T., Geyer-Klingeberg, J., Mendling, J., & Vom Brocke, J. (2022). Creating business value with process mining. *The Journal of Strategic Information Systems*, *31*(4), 101745.

Bapuji, H., Hora, M., Saeed, A., & Turner, S. (2019). How understanding-based redesign influences the pattern of actions and effectiveness of routines. *Journal of Management*, *45*(5), 2132–2162.

Barbon, S., Tavares, G. M., Ceravolo, P., & Damiani, E. (2017). *A framework for trace clustering and concept-drift detection in event streams* (Vol. 2016, pp. 153–154).

Bartik, T. J. (1991). *Who benefits from state and local economic development policies?*

Bartl, E., Rezankova, H., & Sobisek, L. (2011). Comparison of classical dimensionality reduction methods with novel approach based on formal concept analysis. *International Conference on Rough Sets and Knowledge Technology*, 26–35.

Baum, J. A., & Singh, J. V. (1994). Organizational hierarchies and evolutionary processes: Some reflections on a theory of organizational evolution. *Evolutionary Dynamics of Organizations*, *3*, 20.

Becker, M., & Buchkremer, R. (2019). A practical process mining approach for compliance management. *Journal of Financial Regulation and Compliance*, *27*(4), 464–478.

Bell, B. S., & Kozlowski, S. W. (2011). Collective failure: The emergence, consequences, and management of errors in teams. In *Errors in organizations* (pp. 128–156). Routledge.

Benner-Wickner, M., Book, M., Brueckmann, T., & Gruhn, V. (2014). Examining case management demand using event log complexity metrics. *18th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations*, 108–115.

Berente, N., Seidel, S., & Safadi, H. (2019). Research commentary—data-driven computationally intensive theory development. *Information Systems Research*, *30*(1), 50–64.

Bergmann, R., Müller, G., & Wittkowsky, D. (2013). Workflow clustering using semantic similarity measures. *Annual Conference on Artificial Intelligence*, 13–24.

Berkhin, P. (2006). A survey of clustering data mining techniques. In *Grouping multidimensional data* (pp. 25–71). Springer.

Bogarín, A., Romero, C., Cerezo, R., & Sánchez-Santillán, M. (2014). Clustering for improving educational process mining. *ACM International Conference Proceeding Series*, 11–15.

Boltenhagen, M., Chatain, T., & Carmona, J. (2019). *Generalized alignment-based trace clustering of process behavior: Vols. 11522 LNCS* (pp. 237–257). Springer Verlag.

Bose, J. C., Mans, R., & Van der Aalst, W. M. (2013). Wanna improve process mining results? *Symposium on Computational Intelligence and Data Mining*, 127–134.

Bose, J. C., & Van der Aalst, W. M. (2009a). Abstractions in process mining: A taxonomy of patterns. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *5701 LNCS*, 159–175.

Bose, J. C., & Van der Aalst, W. M. (2009b). Context aware trace clustering: Towards improving process mining results. *Proceedings of the 2009 SIAM International Conference on Data Mining*, 401–412.

Bose, J. C., & Van der Aalst, W. M. (2010a). Trace alignment in process mining: Opportunities for process diagnostics. *Business Process Management*, 227–242.

Bose, J. C., & Van der Aalst, W. M. (2010b). Trace clustering based on conserved patterns: Towards achieving better process models. *Lecture Notes in Business Information Processing*, *43 LNBIP*, 170–181.

Bui, H. N., Ha, Q. T., & Nguyen, T. T. (2018). *A novel similarity measure for trace clustering based on normalized google distance* (Special Issue 3; Vol. 15, pp. 341–346).

Bui, H. N., Nguyen, T. T., Nguyen, T. C., & Ha, Q. T. (2018). A new trace clustering algorithm based on context in process mining. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11103 LNAI*, 644–657.

Burger, A., & Moormann, J. (2009). Detecting intrinsic inefficiency on process level: Benchmarking of transactions in banking. In D. Ardagna, M. Mecella, & J. Yang (Eds.), *Business process management workshops* (pp. 146–157). Springer.

Cairns, A. H., Gueni, B., Fhima, M., Cairns, A., David, S., & Khelifa, N. (2014). Towards custom-designed professional training contents and curriculums through educational process mining. *The Fourth International Conference on Advances in Information Mining and Management Towards (IMMM)*, *july*, 53–58.

Card, D. (2001). Immigrant inflows, native outflows, and the local labor market impacts of higher immigration. *Journal of Labor Economics*, *19*(1), 22–64.

Cardoso, J. (2006). Process control-flow complexity metric: An empirical validation. *Proceedings - 2006 IEEE International Conference on Services Computing, SCC 2006*, 167–173.

Cardoso, J., Mendling, J., Neumann, G., & Reijers, H. A. (2006). A discourse on complexity of process models. *International Conference on Business Process Management*, 117–128.

Carmona, J., Dongen, B. van, & Weidlich, M. (2022). Conformance checking: Foundations, milestones and challenges. In *Process mining handbook* (pp. 155–190). Springer.

Carroll, J. S., Christianson, M. K., Frese, M., Lei, Z., Naveh, E., & Vogus, T. J. (2021). FROM THE EDITORS—Errors in Organizations: New Frontiers in Research and Practice. *Academy of Management Discoveries*, *7*(4), 451–460.

Ceravolo, P., Damiani, E., Torabi, M., & Barbon, S. (2017). Toward a new generation of log pre-processing methods for process mining. *Business Process Management*, *297*, 55–70.

Chatain, T., Carmona, J., & Dongen, B. van. (2017). Alignment-based trace clustering. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10650 LNCS*, 295–308.

Chen, H., Chiang, R. H., & Storey, V. C. (2012). Business intelligence and analytics: From big data to big impact. *MIS Quarterly*, *36*(4), 1165–1188.

Chen, W., Ye, K., Wang, Y., Xu, G., & Xu, C. Z. (2019). How does the workload look like in production cloud? Analysis and clustering of workloads on alibaba cluster trace. *Proceedings of the International Conference on Parallel and Distributed Systems - ICPADS*, *2018-Decem*, 102–109.

Cohen, M. D. (2012). *Perceiving and remembering routine action: Fundamental micro-level origins.*

Cohen, M. D., Burkhart, R., Dosi, G., Egidi, M., Marengo, L., Warglien, M., & Winter, S. (1996). Routines and other recurring action patterns of organizations: Contemporary research issues. *Industrial and Corporate Change*, *5*(3), 653–698.

Cragg, J. G., & Donald, S. G. (1993). Testing identifiability and specification in instrumental variable models. *Econometric Theory*, *9*(2), 222–240.

Cross, K., & Lynch, R. (1988). The "SMART" way to define and sustain success. *National Productivity Review*, *8*(1), 23–33.

Cui, R., Li, M., & Li, Q. (2020). Value of high-quality logistics: Evidence from a clash between SF express and alibaba. *Management Science*, *66*(9), 3879–3902.

Cuzzocrea, A., Folino, F., Guarascio, M., & Pontieri, L. (2018). Deviance-aware discovery of high-quality process models. *International Journal on Artificial Intelligence Tools*, *27*(7).

Damaso, K., Williams, P., & Heathcote, A. (2020). Evidence for different types of errors being associated with different types of post-error changes. *Psychonomic Bulletin & Review*, *27*(3), 435–440.

Danielmeier, C., & Ullsperger, M. (2011). Post-error adjustments. *Frontiers in Psychology*, *2*, 1–10.

Darmawan, H., Sarno, R., Ahmadiyah, A. S., Sungkono, K. R., & Wahyuni, C. S. (2018). Anomaly detection based on control-flow pattern of parallel business processes. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, *16*(6), 2809–2816.

Davenport, T. H., Short, J. E., et al. (1990). *The new industrial engineering: Information technology and business process redesign.*

Davis, P., & Peri, T. (2002). Measuring organizational efficiency and effectiveness. *Journal of Management Research*, *2*(2).

De Koninck, P., Broucke, S. vanden, & De Weerdt, J. (2018). act2vec, trace2vec, log2vec, and model2vec: Representation learning for business processes. *International Conference on Business Process Management*, 305–321.

De Koninck, P., & De Weerdt, J. (2017). Multi-objective trace clustering: Finding more balanced solutions. *Lecture Notes in Business Information Processing*, *281*, 49–60.

De Koninck, P., & De Weerdt, J. (2019). Scalable mixed-paradigm trace clustering using super-instances. *Proceedings - 2019 International Conference on Process Mining, ICPM 2019*, 17–24.

De Koninck, P., De Weerdt, J., & Broucke, S. vanden. (2017). *Explaining clusterings of process instances* (3; Vol. 31, pp. 774–808). Springer New York LLC.

De Koninck, P., Nelissen, K., Baesens, B., Vanden Broucke, S., Snoeck, M., & De Weerdt, J. (2017). An approach for incorporating expert knowledge in trace clustering. *Lecture Notes*

in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), *10253 LNCS*, 561–576.
De Leoni, M., Van der Aalst, W. M., & Dees, M. (2016). A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs. *Information Systems*, *56*, 235–257.
De Medeiros, A. A., Pedrinaci, C., Van der Aalst, W. M., Domingue, J., Song, M., Rozinat, A., Norton, B., & Cabral, L. (2007). An outlook on semantic business process mining and monitoring. *On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops: OTM Confederated International Workshops and Posters, AWeSOMe, CAMS, OTM Academy Doctoral Consortium, MONET, OnToContent, ORM, PerSys, PPN, RDDS, SSWS, and SWWS 2007, Vilamoura, Portugal, November 25-30, 2007, Proceedings, Part II*, 1244–1255.
De Weerdt, J., Broucke, S. K. vanden, Vanthienen, J., & Baesens, B. (2012). Leveraging process discovery with trace clustering and text mining for intelligent analysis of incident management processes. *2012 IEEE Congress on Evolutionary Computation*, 1–8.
De Weerdt, J., Vanden Broucke, S., Vanthienen, J., & Baesens, B. (2013). Active trace clustering for improved process discovery. *IEEE Transactions on Knowledge and Data Engineering*, *25*(12), 2708–2720.
Delias, P., Doumpos, M., Grigoroudis, E., Manolitzas, P., & Matsatsinis, N. (2015). Supporting healthcare management decisions via robust clustering of event logs. *Knowledge-Based Systems*, *84*, 203–213.
Delias, P., Doumpos, M., Grigoroudis, E., & Matsatsinis, N. (2019). A non-compensatory approach for trace clustering. *International Transactions in Operational Research*, *26*(5), 1828–1846.
del-Río-Ortega, A., Resinas, M., Cabanillas, C., & Ruiz-Cortés, A. (2013). On the definition and design-time analysis of process performance indicators. *Information Systems*, *38*(4), 470–490.
del-Río-Ortega, A., Resinas, M., Durán, A., Bernárdez, B., Ruiz-Cortés, A., & Toro, M. (2019). Visual PPINOT: A graphical notation for process performance indicators. *Business & Information Systems Engineering*, *61*(2), 137–161.
Deming, W. E. (2018). *Out of the crisis, reissue*. MIT press.
Deokar, A. V., & Tao, J. (2015). Semantics-based event log aggregation for process mining and analytics. *Information Systems Frontiers*, *17*(6), 1209–1226.
Desai, V., & Madsen, P. (2022). Take your time? How activity timing affects organizational learning and performance outcomes. *Organization Science*, *33*(5), 1707–1723.
Di Francescomarino, C., Dumas, M., Maggi, F. M., & Teinemaa, I. (2016). Clustering-based predictive process monitoring. *IEEE Transactions on Services Computing*, *12*(6), 896–909.
Diamantini, C., Genga, L., & Potena, D. (2016). Behavioral process mining for unstructured processes. *Journal of Intelligent Information Systems*, *47*(1), 5–32.
Dobrota, M., Delibašic, B., & Delias, P. (2016). A skiing trace clustering model for injury risk assessment. *International Journal of Decision Support System Technology*, *8*(1), 56–68.
Dogan, O., Oztaysi, B., & Fernandez-Llatas, C. (2020). Segmentation of indoor customer paths using intuitionistic fuzzy clustering: Process mining visualization. *Journal of Intelligent and Fuzzy Systems*, *38*(1), 675–684.
Dohmen, A., & Moormann, J. (2010). Identifying drivers of inefficiency in business processes: A DEA and data mining perspective. In *Enterprise, business-process and information systems modeling* (pp. 120–132). Springer.
Dongen, B. van. (2019). *Dataset BPI challenge 2019*. 4TU Centre for Research Data. https://doi.org/10.4121/uuid:d06aff4b-79f0-45e6-8ec8-e19730c248f1
Dongen, B. van. (2020). *Dataset BPI challenge 2020*. 4TU Centre for Research Data. https://doi.org/10.4121/uuid:ea03d361-a7cd-4f5e-83d8-5fbdf0362550

Duma, D., & Aringhieri, R. (2017). Mining the patient flow through an emergency department to deal with overcrowding. *Springer Proceedings in Mathematics and Statistics*, *210*, 49–59.

Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. (2018). *Fundamentals of Business Process Management* (Vol. 2). Springer.

Dumas, M., Van der Aalst, W. M., & Ter Hofstede, A. H. (2005). *Process-aware information systems: Bridging people and software through process technology.* John Wiley & Sons.

Durante, R., & Zhuravskaya, E. (2018). Attack when the world is not watching? US news and the israeli-palestinian conflict. *Journal of Political Economy*, *126*(3), 1085–1133.

Eggers, J., & Hein, A. (2020). Turning big data into value: A literature review on business value realization from process mining. *ECIS*.

Ekanayake, C. C., Dumas, M., García-Bañuelos, L., & La Rosa, M. (2013). Slice, mine and dice: Complexity-aware automated discovery of business process models. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8094 LNCS*, 49–64.

Elbashir, M. Z., Collier, P. A., & Davern, M. J. (2008). Measuring the effects of business intelligence systems: The relationship between business process and organizational performance. *International Journal of Accounting Information Systems*, *9*(3), 135–153.

El-Wafi, K. (2020). Siemens: Process mining for operational efficiency in Purchase2Pay. In *Process mining in action* (pp. 75–96). Springer.

Evermann, J., Thaler, T., & Fettke, P. (2016). Clustering traces using sequence alignment. *Lecture Notes in Business Information Processing*, *256*, 179–190.

Fani Sani, M., Van der Aalst, W. M., Bolt, A., & García-Algarra, J. (2017). Subgroup discovery in process mining. *Business Information Systems*, *288*, 237–252.

Fani Sani, M., Zelst, S. J. van, & Aalst, W. M. van der. (2018). Applying sequence mining for outlier detection in process mining. *OTM Confederated International Conferences" on the Move to Meaningful Internet Systems"*, 98–116.

Fani Sani, M., Zelst, S. van, & Van der Aalst, W. M. (2019). Repairing outlier behaviour in event logs using contextual behaviour. *Enterprise Modelling and Information Systems Architectures*, *14*.

Feldman, M. S. (2000). Organizational routines as a source of continuous change. *Organization Science*, *11*(6), 611–629.

Feldman, M. S. (2016). Past, present, and future. *Organizational Routines: How They Are Created, Maintained, and Changed*, 23–46.

Feldman, M. S., & Pentland, B. T. (2003). Reconceptualizing organizational routines as a source of flexibility and change. *Administrative Science Quarterly*, *48*(1), 94–118.

Feldman, M. S., Pentland, B. T., D'Adderio, L., & Lazaric, N. (2016). *Beyond routines as things: Introduction to the special issue on routine dynamics.* INFORMS.

Felin, T., Foss, N. J., Heimeriks, K. H., & Madsen, T. L. (2012). Microfoundations of routines and capabilities: Individuals, processes, and structure. *Journal of Management Studies*, *49*(8), 1351–1374.

Ferilli, S., & Angelastro, S. (2020). Efficient declarative-based process mining using an enhanced framework. In *Studies in computational intelligence* (Vol. 880, pp. 121–136).

Ferreira, D. R. (2009). Applied sequence clustering techniques for process mining. In *Handbook of research on business process modeling* (pp. 481–502). IGI Global.

Foerderer, J., & Schuetz, S. W. (2022). Data breach announcements and stock market reactions: A matter of timing? *Management Science*, *68*(10), 7298–7322.

Folino, F., Guarascio, M., & Pontieri, L. (2014). Mining predictive process models out of low-level multidimensional logs. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8484 LNCS*, 533–547.

Folino, F., Guarascio, M., & Pontieri, L. (2015). *Mining multi-variant process models from low-level logs* (Vol. 208, pp. 165–177). Springer Verlag.

Folino, F., Guarascio, M., & Pontieri, L. (2017). A descriptive clustering approach to the analysis of quantitative business-process deviances. *Proceedings of the ACM Symposium on Applied Computing*, *Part F1280*, 765–770.

Frese, M., & Keith, N. (2015). Action errors, error management, and learning in organizations. *Annual Review of Psychology*, *66*, 661–687.

Friberg, R., & Sanctuary, M. (2017). The effect of retail distribution on sales of alcoholic beverages. *Marketing Science*, *36*(4), 626–641.

García-Bañuelos, L., Dumas, M., La Rosa, M., De Weerdt, J., & Ekanayake, C. C. (2014). *Controlled automated discovery of collections of business process models* (Vol. 46, pp. 85–101).

Gebauer, J., & Schober, F. (2006). Information system flexibility and the cost efficiency of business processes. *Journal of the Association for Information Systems*, *7*(3), 8.

Genga, L., Potena, D., Chiorrini, A., Diamantini, C., & Zannone, N. (2020). A latitudinal study on the use of sequential and concurrency patterns in deviance mining. *Studies in Computational Intelligence*, *880*, 103–119.

Ghasemi, M., & Amyot, D. (2016). Process mining in healthcare: A systematised literature review. *International Journal of Electronic Healthcare*, *9*(1), 60–88.

Giardili, S., Ramdas, K., & Williams, J. W. (2022). Leadership and productivity: A study of US automobile assembly plants. *Management Science*.

Goh, J. M., Gao, G., & Agarwal, R. (2011). Evolving work routines: Adaptive routinization of information technology in healthcare. *Information Systems Research*, *22*(3), 565–585.

Golder, S. A., & Macy, M. W. (2014). Digital Footprints: Opportunities and Challenges for Online Social Research. *Annual Review of Sociology*, *40*(1), 129–152.

González, L. S., Rubio, F. G., González, F. R., & Velthuis, M. P. (2010). Measurement in business processes: A systematic review. *Business Process Management Journal*, *16*(1), 114–134.

Graybiel, A. M. et al. (2008). Habits, rituals, and the evaluative brain. *Annual Review of Neuroscience*, *31*(1), 359–387.

Greco, G., Guzzo, A., Pontieri, L., & Sacca, D. (2004). Mining expressive process models by clustering workflow traces. *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 52–62.

Greco, G., Guzzo, A., Pontieri, L., & Saccà, D. (2006). *Discovering expressive process models by clustering log traces* (8; Vol. 18, pp. 1010–1027).

Grisold, T., Wurm, B., Mendling, J., & Vom Brocke, J. (2020). Using process mining to support theorizing about change in organizations. *Proceedings of the 53rd Hawaii International Conference on System Sciences*.

Guenther, C. W., Rozinat, A., & Van der Aalst, W. M. (2010). Activity mining by global trace segmentation. *Lecture Notes in Business Information Processing*, *43 LNBIP*, 128–139.

Günther, C. (2009). *Process mining in flexible environments* [PhD thesis]. Eindhoven University of Technology.

Günther, C., Rozinat, a., Aalst, W. van der, & Uden, K. van. (2008). *Monitoring deployed application usage with process mining* (pp. 1–8).

Ha, Q. T., Bui, H. N., & Nguyen, T. T. (2016). A trace clustering solution based on using the distance graph model. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9875 LNCS*, 313–322.

Harmon, P. (2019). *Business process change: A business process management guide for managers and process professionals*. Morgan Kaufmann.

Haunschild, P. R., Polidoro Jr, F., & Chandler, D. (2015). Organizational oscillation between learning and forgetting: The dual role of serious errors. *Organization Science*, *26*(6), 1682–1701.

Helm, E., Lin, A. M., Baumgartner, D., Lin, A. C., & Küng, J. (2020). Towards the use of standardized terms in clinical case studies for process mining in healthcare †. *International Journal of Environmental Research and Public Health Article*.

Hodgson, G. M. (2003). The Mystery of the Routine: The Darwinian Destiny of An Evolutionary Theory of Economic Change. *Revue Économique*, *54*(2), 355–384.

Höhenberger, S., & Delfmann, P. (2015). Supporting business process improvement through business process weakness pattern collections. *Wirtschaftsinformatik*, *2015*, 378–392.

Hompes, B. F. A., Buijs, J. C. A. M., Van der Aalst, W. M., Dixit, P. M., & Buurman, J. (2015a). *Detecting change in procebes using comparative trace clustering* (Vol. 1527, pp. 95–108).

Hompes, B. F. A., Buijs, J. C. A. M., Van der Aalst, W. M., Dixit, P. M., & Buurman, J. (2015b). *Discovering deviating cases and process variants using trace clustering*.

Hompes, B., Buijs, J., & Van der Aalst, W. M. (2016). A generic framework for context-aware process performance analysis. In C. Debruyne et al. (Eds.), *On the move to meaningful internet systems* (pp. 300–317). Springer.

Huang, Z., Dong, W., Duan, H., & Li, H. (2014). Similarity measure between patient traces for clinical pathway analysis: Problem, method, and applications. *IEEE Journal of Biomedical and Health Informatics*, *18*(1), 4–14.

Huang, Z., Dong, W., Wang, F., & Duan, H. (2015). Medical inpatient journey modeling and clustering: A bayesian hidden markov model based approach. *AMIA ... Annual Symposium Proceedings. AMIA Symposium*, *2015*, 649–658.

ISO/DIS 9001:redline:2014(e). (2014). https://www.iso.org/files/live/sites/isoorg/files/archive/pdf/en/iso9001-2015-process-appr.pdf

Jablonski, S., Röglinger, M., Schönig, S., & Wyrtki, K. M. (2019). Multi-perspective clustering of process execution traces. *Enterprise Modelling and Information Systems Architectures (EMISAJ)*, *14*, 2–1.

Jin, W., Liu, T., Zheng, Q., Cui, D., & Cai, Y. (2018). Functionality-oriented microservice extraction based on execution trace clustering. *Proceedings - 2018 IEEE International Conference on Web Services, ICWS 2018 - Part of the 2018 IEEE World Congress on Services*, 211–218.

Joseph, J., Rhee, L., & Wilson, A. J. (2022). Corporate hierarchy and organizational learning: Member turnover, code change, and innovation in the multiunit firm. *Organization Science*.

Kanj, S., Brüls, T., & Gazut, S. (2018). Shared nearest neighbor clustering in a locality sensitive hashing framework. *Journal of Computational Biology*, *25*(2), 236–250.

Kaplan, R., & Norton, D. (1998). Putting the balanced scorecard to work. *The Economic Impact of Knowledge*, *27*(4), 315–324.

Kaplan, R., & Norton, D. (2001). *The strategy-focused organization: How balanced scorecard companies thrive in the new business environment*. Harvard Business Press.

Kherbouche, M., Laga, N., & Masse, P. (2017, February). Towards a better assessment of event logs quality. *Symposium Series on Computational Intelligence*.

Kleibergen, F., & Paap, R. (2006). Generalized reduced rank tests using the singular value decomposition. *Journal of Econometrics*, *133*(1), 97–126.

Koninck, P. D., & Weerdt, J. D. (2017). Similarity-based approaches for determining the number of trace clusters in process discovery. In *Transactions on petri nets and other models of concurrency XII* (pp. 19–42).

Koschmider, A. (2017). Clustering event traces by behavioral similarity. *International Conference on Conceptual Modeling*, 36–42.

Koschmider, A., & Moreira, D. S. V. (2018). Change detection in event logs by clustering. *OTM Confederated International Conferences" on the Move to Meaningful Internet Systems"*, 643–660.

Koster, J., & Aven, B. (2018). The effects of individual status and group performance on network ties among teammates in the national basketball association. *PloS One*, *13*(4), e0196013.

Kueng, P. (2000). Process performance measurement system: A tool to support process-based organizations. *Total Quality Management*, *11*(1), 67–85.

La Rosa, M., Dumas, M., Ekanayake, C. C., García-Bañuelos, L., Recker, J., & Ter Hofstede, A. H. M. (2015). Detecting approximate clones in business process model repositories. *Information Systems*, *49*, 102–125.

Landau, S., Leese, M., Stahl, D., & Everitt, B. S. (2011). *Cluster analysis*. John Wiley & Sons.

Larkin, I., Pierce, L., Shalvi, S., & Tenbrunsel, A. (2021). The opportunities and challenges of behavioral field research on misconduct. *Organizational Behavior and Human Decision Processes*, *166*, 1–8.

Latva-Koivisto, A. (2001). *Finding a complexity measure for business process models*. Helsinki University of Technology.

Lazaric, N. (2000). The role of routines, rules and habits in collective learning: Some epistemological and ontological considerations. *European Journal of Economic and Social Systems*, *14*(2), 157–171.

Lazaric, N., Denis, B., et al. (2001). *How and why routines change: Some lessons from the articulation of knowledge with ISO 9002 implementation in the food industry.*

Lee, D., Park, J., Pulshashi, I. R., & Bae, H. (2017). Clustering and operation analysis for assembly blocks using process mining in shipbuilding industry. *Lecture Notes in Business Information Processing*, *159*, 67–80.

Lee, S., Kim, B., Huh, M., Cho, S., Park, S., & Lee, D. (2013). Mining transportation logs for understanding the after-assembly block manufacturing process in the shipbuilding industry. *Expert Systems with Applications*, *40*(1), 83–95.

Lei, Z., Naveh, E., & Novikov, Z. (2016). Errors in organizations: An integrative review via level of analysis, temporal dynamism, and priority lenses. *Journal of Management*, *42*(5), 1315–1343.

Levenshtein, V. I. et al. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, *10*, 707–710.

Leyer, M. (2011). Towards a context-aware analysis of business process performance. *Proceedings of the 15th Pacific Asia Conference of Information Systems.*

Li, C., Reichert, M., & Wombacher, A. (2010). The MinAdept clustering approach for discovering reference process models out of process variants. *International Journal of Cooperative Information Systems*, *19*(03n04), 159–203.

Li, J., Bose, J. C., & Van der Aalst, W. M. (2011). Mining context-dependent and interactive business process maps using execution patterns. *Lecture Notes in Business Information Processing*, *66 LNBIP*, 109–121.

Liu, C.-W., Gao, G., & Agarwal, R. (2019). Unraveling the "social" in social norms: The conditioning effect of user connectivity. *Information Systems Research*, *30*(4), 1272–1295.

Louis, M. R., & Sutton, R. I. (1991). Switching cognitive gears: From habits of mind to active thinking. *Human Relations*, *44*(1), 55–76.

Lu, X., Tabatabaei, S. A., Hoogendoorn, M., & Reijers, H. A. (2019). Trace clustering on very large event data in healthcare using frequent sequence patterns. *Business Process Management: 17th International Conference, BPM 2019, Vienna, Austria, September 1–6, 2019, Proceedings 17*, 198–215.

Luengo, D., & Sepúlveda, M. (2012). Applying clustering in process mining to find different versions of a business process that changes over time. *Lecture Notes in Business Information Processing*, *99 LNBIP*, 153–158.

Madsen, P. M., & Desai, V. (2010). Failing to learn? The effects of failure and success on organizational learning in the global orbital launch vehicle industry. *Academy of Management Journal*, *53*(3), 451–476.

Mahendrawathi, E., Arsad, N., Astuti, H. M., Kusumawardani, R. P., & Utami, R. A. (2018). Analysis of production planning in a global manufacturing company with process mining. *Journal of Enterprise Information Management*.

Mahringer, C. A., & Pentland, B. T. (2021). Sequence analysis in routine dynamics. *Cambridge Handbook of Routine Dynamics*, 172–183.

Maita, A. R. C., Martins, L. C., Paz, C. R. L., Peres, S. M., & Fantinato, M. (2015). Process mining through artificial neural networks and support vector machines: A systematic literature review. *Business Process Management Journal*, *21*(6), 1391–1415.

Makary, M. A., & Daniel, M. (2016). Medical error—the third leading cause of death in the US. *BMJ*, *353*, i2139.

Mans, R., Reijers, H., Berends, H., Bandara, W., & Prince, R. (2013). Business process mining success. In R. Connolly, E. van Heck, & S. Spiekermann (Eds.), *Proceedings of the 21st European Conference on Information Systems* (pp. 1–13). Association for Information Systems (AIS).

Mans, R., Schonenberg, M., Song, M., Van der Aalst, W. M., & Bakker, P. J. (2009). Application of process mining in healthcare–a case study in a dutch hospital. *Biomedical Engineering Systems and Technologies: International Joint Conference, BIOSTEC 2008 Funchal, Madeira, Portugal, January 28-31, 2008 Revised Selected Papers 1*, 425–438.

Márquez-Chamorro, A. E., Resinas, M., & Ruiz-Cortés, A. (2017). Predictive monitoring of business processes: A survey. *IEEE Transactions on Services Computing*, *11*(6), 962–977.

Martin, N., Depaire, B., & Caris, A. (2016). The use of process mining in business process simulation model construction. *Business & Information Systems Engineering*, *58*(1), 73–87.

McCormack, K. P., & Johnson, W. C. (2001). *Business process orientation: Gaining the e-business competitive advantage*. Crc Press.

Medeiros, A. K. A. de, Guzzo, A., Greco, G., Van der Aalst, W. M., Weijters, A., Dongen, B. F. van, & Sacca, D. (2007). Process mining based on clustering: A quest for precision. *International Conference on Business Process Management*, 17–29.

Meincheim, A., Garcia, C. D. S., Nievola, J. C., & Scalabrin, E. E. (2018). Combining process mining with trace clustering: Manufacturing shop floor process-an applied case. *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, *2017-Novem*, 498–505.

Melville, N., Kraemer, K., & Gurbaxani, V. (2004). Information technology and organizational performance: An integrative model of IT business value. *MIS Quarterly*, 283–322.

Mendling, J. (2007). *Detection and prediction of errors in EPC business process models* [PhD thesis]. Vienna University of Business Administration.

Mendling, J., Neumann, G., & Van der Aalst, W. M. (2007a). On the correlation between process model metrics and errors. In J. Grundy, A. Laender, J. Roddick, S. Hartmann, & L. Maciaszek (Eds.), *Challenges in conceptual modelling, tutorials posters panels and industrial contributions to the 26th international conference on conceptual modeling* (pp. 173–178). Australian Computer Society. https://eprints.qut.edu.au/13958/

Mendling, J., Neumann, G., & Van der Aalst, W. M. (2007b). Understanding the occurrence of errors in process models based on metrics. *OTM Confederated International Conferences" on the Move to Meaningful Internet Systems"*, 113–130.

Mendling, J., Reijers, H. A., & Cardoso, J. (2007). What makes process models understandable? *International Conference on Business Process Management*, 48–63.

Montani, S., & Leonardi, G. (2014). Retrieval and clustering for supporting business process adjustment and analysis. *Information Systems*, *40*, 128–141.

Montani, S., Leonardi, G., Striani, M., Quaglini, S., & Cavallini, A. (2017). Multi-level abstraction for trace comparison and process discovery. *Expert Systems with Applications*, *81*, 398–409.

Murillas, E. de, Reijers, H. A., & Van der Aalst, W. M. (2020). Case notion discovery and recommendation: Automated event log building on databases. *Knowledge and Information Systems*, *62*(7), 2539–2575.

Najjar, A., Reinharz, D., Girouard, C., & Gagné, C. (2018). A two-step approach for mining patient treatment pathways in administrative healthcare databases. *Artificial Intelligence in Medicine*, *87*, 34–48.

Neely, A., Richards, H., Platts, K., Bourne, M., Mills, J., Gregory, M., & Kennerley, M. (2000). Performance measurement system design: Developing and testing a process-based approach. *International Journal of Operations & Production Management*, *20*(10), 1119–1145.

Nguyen, H., Dumas, M., Hofstede, A. H. ter, La Rosa, M., & Maggi, F. M. (2019). Stage-based discovery of business process models from event logs. *Information Systems*, *84*, 214–237.

Nguyen, P., Slominski, A., Muthusamy, V., Ishakian, V., & Nahrstedt, K. (2016). *Process trace clustering: A heterogeneous information network approach* (pp. 279–287). Society for Industrial; Applied Mathematics Publications.

Notebaert, W., Houtman, F., Van Opstal, F., Gevers, W., Fias, W., & Verguts, T. (2009). Post-error slowing: An orienting account. *Cognition*, *111*(2), 275–279.

Oliver, N., Calvard, T., & Potočnik, K. (2017). Cognition, technology, and organizational limits: Lessons from the Air France 447 disaster. *Organization Science*, *28*(4), 729–743.

Ordoñez, A., Ordoñez, H., Corrales, J. C., Cobos, C., Wives, L. K., & Thom, L. H. (2017). Grouping of business processes models based on an incremental clustering algorithm using fuzzy similarity and multimodal search. *Expert Systems with Applications*, *67*, 163–177.

Pentland, B. T., & Feldman, M. S. (2005). Organizational routines as a unit of analysis. *Industrial and Corporate Change*, *14*(5), 793–815.

Pentland, B. T., & Feldman, M. S. (2008). Designing routines: On the folly of designing artifacts, while hoping for patterns of action. *Information and Organization*, *18*(4), 235–250.

Pentland, B. T., Haerem, T., & Hillison, D. (2011). The (n) ever-changing world: Stability and change in organizational routines. *Organization Science*, *22*(6), 1369–1383.

Pentland, B. T., Liu, P., Kremser, W., & Haerem, T. (2020). The dynamics of drift in digitized processes. *MIS Quarterly*, *44*(1), 19–47.

Pentland, B. T., Mahringer, C. A., Dittrich, K., Feldman, M. S., & Wolf, J. R. (2020). Process multiplicity and process dynamics: Weaving the space of possible paths. *Organization Theory*, *1*(3).

Pentland, B. T., Recker, J., Wolf, J. R., & Wyner, G. (2020). Bringing context inside process research with digital trace data. *Journal of the Association for Information Systems*, *21*(5), 5.

Pentland, B. T., & Rueter, H. H. (1994). Organizational routines as grammars of action. *Administrative Science Quarterly*, 484–510.

Pentland, B. T., Vaast, E., & Wolf, J. R. (2021). Theorizing Process Dynamics with Directed Graphs: A Diachronic Analysis of Digital Trace Data. *MIS Quarterly*, *45*(2), 967–983.

Prathama, F., Yahya, B. N., Harjono, D. D., & Mahendrawathi, E. R. (2019). Trace clustering exploration for detecting sudden drift: A case study in logistic process. *Procedia Computer Science*, *161*, 1122–1130.

Priyambada, S. A., Er, M., & Yahya, B. N. (2018). Curriculum assessment of higher educational institution using trace-segmented clustering. *Curriculum Assessment of Higher Educational Institution Using Trace-Segmented Clustering*, *20*(1), 33–48.

Rahmandad, H., Repenning, N., & Sterman, J. (2009). Effects of feedback delay on learning. *System Dynamics Review*, *25*(4), 309–338.

Razi, M. A., Sungkono, K. R., Sarno, R., & Wahyuni, C. S. (2019). Detecting bottleneck and fraud in agile development by using petri net performance and trace clustering. *Proceedings - 2019 International Seminar on Application for Technology of Information and Communication: Industry 4.0: Retrospect, Prospect, and Challenges, iSemantic 2019*, 214–218.

Rebuge, A., & Ferreira, D. R. (2012). Business process analysis in healthcare environments: A methodology based on process mining. *Information Systems*, *37*(2), 99–116.
Reijers, H. A., Mendling, J., & Dijkman, R. M. (2011). Human and automatic modularizations of process models to enhance their comprehension. *Information Systems*, *36*(5), 881–897.
Reinkemeyer, L. (2020). *Process mining in action*. Springer.
Ribeiro, J., Carmona, J., Misir, M., & Sebag, M. (2014). A recommender system for process discovery. *Business Process Management*, 67–83.
Richetti, P. H. P., Baião, F. A., & Santoro, F. M. (2014). Declarative process mining: Reducing discovered models complexity by pre-processing event logs. *Business Process Management*, *8659 LNCS*, 400–407.
Richter, F., Wahl, F., Sydorova, A., & Seidl, T. (2019). K-process: Model-conformance-based clustering of process instances. *LWDA*, 161–172.
Richter, F., Zellner, L., Sontheim, J., & Seidl, T. (2019). Model-aware clustering of non-conforming traces. *OTM Confederated International Conferences on the Move to Meaningful Internet Systems*, 193–200.
Rosenberg, A., Chase, G., Omar, R., Taylor, J., & Rosing, M. von. (2011). *Applying real-world BPM in an SAP environment*. Galileo Press.
Rowlson, M. (2020). Uber: Process mining to optimize customer experience and business performance. In *Process mining in action* (pp. 59–63). Springer.
Sakr, S., Maamar, Z., Awad, A., Benatallah, B., & Van Der Aalst, W. M. (2018). Business process analytics and big data systems: A roadmap to bridge the gap. *IEEE Access*, *6*, 77308–77320.
Sánchez-González, L., García, F., Mendling, J., Ruiz, F., & Piattini, M. (2010). Prediction of business process model quality based on structural metrics. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *6412 LNCS*, 458–463.
Schroder, H. S., & Moser, J. S. (2014). Improving the study of error monitoring with consideration of behavioral performance measures. In *Frontiers in Human Neuroscience* (Vol. 8, p. 178).
Seeliger, A., Nolle, T., & Mühlhäuser, M. (2017). Detecting concept drift in processes using graph metrics on process graphs. *Proceedings of the 9th Conference on Subject-Oriented Business Process Management*, 1–10.
Seeliger, A., Nolle, T., & Mühlhäuser, M. (2018). Finding structure in the unstructured: Hybrid feature set clustering for process discovery. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11080 LNCS*, 288–304.
Seeliger, A., Sánchez Guinea, A., Nolle, T., & Mühlhäuser, M. (2019). ProcessExplorer: Intelligent process mining guidance. *International Conference on Business Process Management*, 216–231.
Segalowitz, S. J., Santesso, D. L., Murphy, T. I., Homan, D., Chantziantoniou, D. K., & Khan, S. (2010). Retest reliability of medial frontal negativities during performance monitoring. *Psychophysiology*, *47*(2), 260–270.
Seret, A., Vanden Broucke, S. K. L. M., Baesens, B., & Vanthienen, J. (2014). A dynamic understanding of customer behavior processes based on clustering and sequence mining. *Expert Systems with Applications*, *41*(10), 4648–4657.
Shahin, A., & Mahbod, M. A. (2007). Prioritization of key performance indicators: An integration of analytical hierarchy process and goal setting. *International Journal of Productivity and Performance Management*, *56*(3), 226–240.
Sitkin, S. B. (1992). Learning through failure: The strategy of small losses. *Research in Organizational Behavior*, *14*, 231–266.

Song, M. S., CW, G., WMP, A., & Jung, J. Y. (2008). *Improving process mining with trace clustering* (4; Vol. 34, pp. 460–469).

Song, M., Guenther, C. W., & Van der Aalst, W. M. (2009). *Trace clustering in process mining: Vols. 17 LNBIP* (pp. 109–120). Springer Verlag.

Song, M., Yang, H., Siadat, S. H., & Pechenizkiy, M. (2013). *A comparative study of dimensionality reduction techniques to enhance trace clustering performances* (9; Vol. 40, pp. 3722–3737).

Stern, Z., Katz-Navon, T., & Naveh, E. (2008). The influence of situational learning orientation, autonomy, and voice on error making: The case of resident physicians. *Management Science*, *54*(9), 1553–1564.

Stock, J., & Yogo, M. (2005). Asymptotic distributions of instrumental variables statistics with many instruments. *Identification and Inference for Econometric Models: Essays in Honor of Thomas Rothenberg*, *6*, 109–120.

Stocker, T. (2012). Time-based trace clustering for evolution-aware security audits. *Lecture Notes in Business Information Processing*, *100 LNBIP*(PART 2), 471–476.

Stuart, E. A., King, G., Imai, K., & Ho, D. (2011). MatchIt: Nonparametric preprocessing for parametric causal inference. *Journal of Statistical Software*.

Sun, Y., Dong, X., & McIntyre, S. (2017). Motivation of user-generated content: Social connectedness moderates the effects of monetary rewards. *Marketing Science*, *36*(3), 329–337.

Suriadi, S., Andrews, R., Hofstede, A. H. ter, & Wynn, M. T. (2017). Event log imperfection patterns for process mining: Towards a systematic approach to cleaning event logs. *Information Systems*, *64*, 132–150.

Swennen, M., Janssenswillen, G., Jans, M., Depaire, B., & Vanhoof, K. (2015). Capturing process behavior with log-based process metrics. *SIMPDA*, 141–144.

Tax, N., Sidorova, N., Haakma, R., & Van der Aalst, W. M. (2016). Mining local process models. *Journal of Innovation in Digital Ecosystems*, *3*(2), 183–196.

Taymouri, F., La Rosa, M., Dumas, M., & Maggi, F. M. (2021). Business process variant analysis: Survey and classification. *Knowledge-Based Systems*, *211*, 106557.

Taymouri, F., La Rosa, M., Erfani, S., Bozorgi, Z. D., & Verenich, I. (2020). Predictive business process monitoring via generative adversarial nets: The case of next event prediction. *International Conference on Business Process Management*, 237–256.

Taymouri, F., Rosa, M. L., & Carmona, J. (2020). Business process variant analysis based on mutual fingerprints of event logs. *International Conference on Advanced Information Systems Engineering*, 299–318.

Thaler, T., Ternis, S., Fettke, P., & Loos, P. (2015). A comparative analysis of process instance cluster techniques. *Proceedings Der 12. Internationalen Tagung Wirtschaftsinformatik*, *WI 2015*, 423–437.

Turner, S. F., & Rindova, V. P. (2018). Watching the clock: Action timing, patterning, and routine performance. *Academy of Management Journal*, *61*(4), 1253–1280.

Van den Broucke, S., Delvaux, C., Freitas, J., Rogova, T., Vanthienen, J., & Baesens, B. (2014). Uncovering the relationship between event log characteristics and process discovery techniques. *Business Process Management*, 41–53.

Van Den Ingh, L., Eshuis, R., & Gelper, S. (2020). Assessing performance of mined business process variants. *Enterprise Information Systems*.

Van der Aalst, W. M. (2008). *Process-aware information systems: Design, enactment, and analysis.*

Van der Aalst, W. M. (2009). Process-aware information systems: Lessons to be learned from process mining. In *Transactions on petri nets and other models of concurrency II* (pp. 1–26).

Van der Aalst, W. M. (2012a). Process mining: Overview and opportunities. *ACM Transactions on Management Information Systems (TMIS)*, *3*(2), 1–17.

Van der Aalst, W. M. (2012b). Service mining: Using process mining to discover, check, and improve service behavior. *IEEE Transactions on Services Computing*, *6*(4), 525–535.

Van der Aalst, W. M. (2013). Business process management: A comprehensive survey. *ISRN Software Engineering*, *2013*, 507984.

Van der Aalst, W. M. (2016). *Process mining: Data science in action* (Vol. 2). Springer.

Van der Aalst, W. M., Adriansyah, A., Medeiros, A. K. A. de, Arcieri, F., Baier, T., Blickle, T., Bose, J. C., Brand, P. van den, Brandtjen, R., Buijs, J., et al. (2011). Process mining manifesto. *International Conference on Business Process Management*, 169–194.

Van der Aalst, W. M., & Dustdar, S. (2012). Process mining put into context. *Internet Computing*, *16*(1), 82–86.

Van Der Aalst, W. M., Reijers, H. A., Weijters, A. J., Dongen, B. F. van, De Medeiros, A. A., Song, M., & Verbeek, H. (2007). Business process mining: An industrial application. *Information Systems*, *32*(5), 713–732.

Van Dyck, C., Frese, M., Baer, M., & Sonnentag, S. (2005). Organizational error management culture and its impact on performance: A two-study replication. *Journal of Applied Psychology*, *90*(6), 1228.

Van Looy, A., & Shafagatova, A. (2016). Business process performance measurement: A structured literature review of indicators, measures and metrics. *SpringerPlus*, *5*(1), 1–24.

Veiga, G. M., & Ferreira, D. R. (2010). *Understanding spaghetti models with sequence clustering for ProM: Vols. 43 LNBIP* (pp. 92–103). Springer Verlag.

Verenich, I., Dumas, M., La Rosa, M., Maggi, F. M., & Di Francescomarino, C. (2016). Complex symbolic sequence clustering and multiple classifiers for predictive process monitoring. *Business Process Management*, *256*, 218–229.

Wang, C., Zhang, X., & Hann, I.-H. (2018). Socially nudged: A quasi-experimental study of friends' social influence in online product ratings. *Information Systems Research*, *29*(3), 641–655.

Webster, J., & Watson, R. T. (2002). Analyzing the past to prepare for the future: Writing a literature review. *MIS Quarterly*, xiii–xxiii.

Weijters, A., & Aalst, W. M. van der. (2001). Process mining: Discovering workflow models from event-based data. *Belgium-Netherlands Conf. On Artificial Intelligence.*

Winter, S. G., & Nelson, R. R. (1982). An evolutionary theory of economic change. *University of Illinois at Urbana-Champaign's Academy for Entrepreneurial Leadership Historical Research Reference in Entrepreneurship.*

Wood, W. (2017). Habit in personality and social psychology. *Personality and Social Psychology Review*, *21*(4), 389–403.

Wood, W., & Neal, D. T. (2009). The habitual consumer. *Journal of Consumer Psychology*, *19*(4), 579–592.

Wurm, B., Grisold, T., Mendling, J., & Brocke, J. vom. (2021). Business process management and routine dynamics. *Cambridge Handbook of Routine Dynamics*, 513–524.

Yang, S., Dong, X., Sun, L., Zhou, Y., Farneth, R. A., Xiong, H., Burd, R. S., & Marsic, I. (2017). A data-driven process recommender framework. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Part F1296*, 2111–2120.

Yeshchenko, A., Bayomie, D., Gross, S., & Mendling, J. (2020). Visualizing business process evolution. In *Enterprise, business-process and information systems modeling* (pp. 185–192).

Yu Chung Wang, W., Pauleen, D., & Taskin, N. (2022). Enterprise systems, emerging technologies, and the data-driven knowledge organisation. In *Knowledge Management Research & Practice* (1; Vol. 20, pp. 1–13). Taylor & Francis.

Zandkarimi, F., Decker, P., & Rehse, J.-R. (2021). Fig4PM: A library for calculating event log measures. *ICPM Doctoral Consortium and Demo Track*, 27–28.

Zandkarimi, F., Rehse, J.-R., Soudmand, P., & Hoehle, H. (2020). A generic framework for trace clustering in process mining. *2020 2nd International Conference on Process Mining (ICPM)*, 177–184.

Zandkarimi, F., Rennemeier, J., & Rehse, J.-R. (2021). Are we doing things right? An approach to measure process inefficiencies in the control flow. *International Conference on Business Process Management*, 109–125.

Zellmer-Bruhn, M. E. (2003). Interruptive events and team knowledge acquisition. *Management Science*, *49*(4), 514–528.

Zhao, X., Liang, J., & Dang, C. (2019). A stratified sampling based clustering algorithm for large-scale data. *Knowledge-Based Systems*, *163*, 416–428.

Zheng, A., & Casari, A. (2018). *Feature engineering for machine learning: Principles and techniques for data scientists*. O'Reilly Media.

Appendixes

A Appendix A

Table A: List of trace clustering articles in the literature review

Article	Author
Mining expressive process models by clustering workflow traces	Greco et al. (2004)
Discovering expressive process models by clustering log traces	Greco et al. (2006)
Monitoring deployed application usage with process mining	Günther et al. (2008)
Improving process mining with trace clustering	M. S. Song et al. (2008)
Activity mining by global trace segmentation	Guenther et al. (2010)
Abstractions in process mining: A taxonomy of patterns	Bose & Van der Aalst (2009a)
Context aware trace clustering: Towards improving process mining results	Bose & Van der Aalst (2009b)
Trace alignment in process mining: opportunities for process diagnostics	Bose & Van der Aalst (2010a)
Understanding spaghetti models with sequence clustering for ProM	Veiga & Ferreira (2010)
Trace clustering based on conserved patterns: Towards achieving better process models	Bose & Van der Aalst (2010b)
Mining context-dependent and interactive business process maps using execution patterns	J. Li et al. (2011)
Applying clustering in process mining to find different versions of a business process that changes over time	Luengo & Sepúlveda (2012)
Business process analysis in healthcare environments: A methodology based on process mining	Rebuge & Ferreira (2012)
Time-based trace clustering for evolution-aware security audits	Stocker (2012)
Leveraging process discovery with trace clustering and text mining for intelligent analysis of incident management processes	De Weerdt et al. (2012)
Analysis framework using process mining for block movement process in shipyards	De Weerdt et al. (2012)
Similarity measure between patient traces for clinical pathway analysis: problem, method, and applications	Huang et al. (2014)
Slice, mine and dice: Complexity-aware automated discovery of business process models	Ekanayake et al. (2013)
A comparative study of dimensionality reduction techniques to enhance trace clustering performances	Song et al. (2013)
Active trace clustering for improved process discovery	De Weerdt et al. (2013)
Wanna improve process mining results?	Bose et al. (2013)
Clustering for improving educational process mining	Bogarín et al. (2014)
Controlled automated discovery of collections of business process models	García-Bañuelos et al. (2014)
Declarative process mining: reducing discovered models complexity by pre-processing event logs	Richetti et al. (2014)
Mining predictive process models out of low-level multidimensional logs	Folino et al. (2014)
Towards custom-designed professional training contents and curriculums through educational process mining	Cairns et al. (2014)

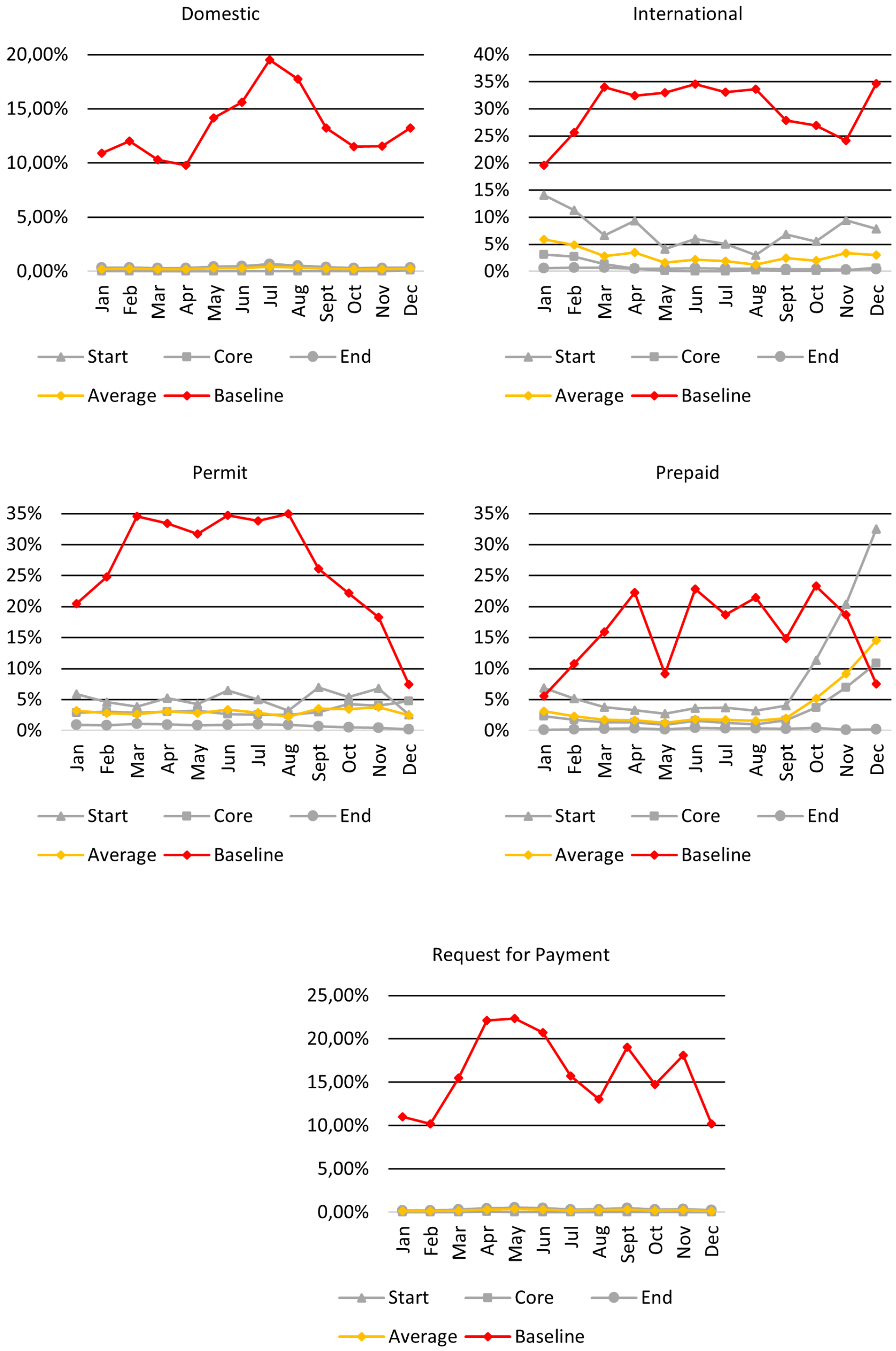
Article	Author
A dynamic understanding of customer behavior processes based on clustering and sequence mining	Seret et al. (2014)
Retrieval and clustering for supporting business process adjustment and analysis	Montani & Leonardi (2014)
Detecting approximate clones in business process model repositories	La Rosa et al. (2015)
Medical inpatient journey modeling and clustering: a Bayesian hidden Markov model based approach	Huang et al. (2015)
Detecting change in procebes using comparative trace clustering	B. F. A. Hompes et al. (2015a)
Semantics-based event log aggregation for process mining and analytics	Deokar & Tao (2015)
Mining multi-variant process models from low-level logs	Folino et al. (2015)
A Comparative Analysis of Process Instance Cluster Techniques	Thaler et al. (2015)
Discovering deviating cases and process variants using trace clustering	B. F. A. Hompes et al. (2015b)
A co-training strategy for multiple view clustering in process mining	Appice & Malerba (2016)
Supporting healthcare management decisions via robust clustering of event logs	Delias et al. (2015)
Complex symbolic sequence clustering and multiple classifiers for predictive process monitoring	Verenich et al. (2016)
Mining local process models	Tax et al. (2016)
Process mining in healthcare: a systematised literature review	Ghasemi & Amyot (2016)
A skiing trace clustering model for injury risk assessment	Dobrota et al. (2016)
Behavioral process mining for unstructured processes	Diamantini et al. (2016)
Process trace clustering: A heterogeneous information network approach	P. Nguyen et al. (2016)
A trace clustering solution based on using the distance graph model	Ha et al. (2016)
Clustering traces using sequence alignment	Evermann et al. (2016)
Clustering-based predictive process monitoring	Di Francescomarino et al. (2016)
A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs	De Leoni et al. (2016)
Grouping of business processes models based on an incremental clustering algorithm using fuzzy similarity and multimodal search	Ordoñez et al. (2017)
Mining the patient flow through an Emergency Department to deal with overcrowding	Duma & Aringhieri (2017)
A descriptive clustering approach to the analysis of quantitative business-process deviances	Folino et al. (2017)
Clustering and operation analysis for assembly blocks using process mining in shipbuilding industry	D. Lee et al. (2017)
Multi-objective trace clustering: finding more balanced solutions	De Koninck & De Weerdt (2017)
A Framework for Trace Clustering and Concept-drift Detection in Event Streams	Barbon et al. (2017)
A novel trace clustering technique based on constrained trace alignment	Wang et al. (2018)

Article	Author
Combining Process Mining with Trace Clustering: Manufacturing Shop Floor Process-An Applied Case	Meincheim et al. (2018)
An approach for incorporating expert knowledge in trace clustering	De Koninck, Nelissen, et al. (2017)
Subgroup discovery in process mining	Fani Sani et al. (2017)
Multi-level abstraction for trace comparison and process discovery	Montani et al. (2017)
Explaining clusterings of process instances	De Koninck, De Weerdt, et al. (2017)
Toward a new generation of log pre-processing methods for process mining	Ceravolo et al. (2017)
Compound trace clustering to generate accurate and simple sub-process models	Sun et al. (2017)
Alignment-based trace clustering	Chatain et al. (2017)
A data-driven process recommender framework	Yang et al. (2017)
Deviance-Aware Discovery of High-Quality Process Models	Cuzzocrea et al. (2018)
Functionality-oriented microservice extraction based on execution trace clustering	Jin et al. (2018)
How does the workload look like in production cloud? analysis and clustering of workloads on alibaba cluster trace	W. Chen et al. (2019)
A novel similarity measure for trace clustering based on normalized google distance.	Bui, Ha, et al. (2018)
A two-step approach for mining patient treatment pathways in administrative healthcare databases	Najjar et al. (2018)
Curriculum Assessment of Higher Educational Institution Using Trace-segmented Clustering	Priyambada et al. (2018)
A New Trace Clustering Algorithm Based on Context in Process Mining	Bui, Nguyen, et al. (2018)
Detecting Bottleneck and Fraud in Agile Development by using Petri net Performance and Trace Clustering	Razi et al. (2019)
Trace Clustering Exploration for Detecting Sudden Drift: A Case Study in Logistic Process	Prathama et al. (2019)
Business Process Variant Analysis: Survey and Classification	Taymouri et al. (2021)
Scalable Mixed-Paradigm Trace Clustering using Super-Instances	De Koninck & De Weerdt (2019)
Generalized alignment-based trace clustering of process behavior	Boltenhagen et al. (2019)
Trace Clustering on Very Large Event Data in Healthcare Using Frequent Sequence Patterns	Lu et al. (2019)
Towards the Use of Standardized Terms in Clinical Case Studies for Process Mining in Healthcare	Helm et al. (2020)
A new method for organizational process model discovery through the analysis of workflows and data exchange networks	Aghabaghery et al. (2020)
Efficient Declarative-Based Process Mining Using an Enhanced Framework	Ferilli & Angelastro (2020)
A latitudinal study on the use of sequential and concurrency patterns in deviance mining	Genga et al. (2020)
Segmentation of indoor customer paths using intuitionistic fuzzy clustering: Process mining visualization	Dogan et al. (2020)
Finding structure in the unstructured: hybrid feature set clustering for process discovery	Seeliger et al. (2018)

Article	Author
Bringing context inside process research with digital trace data	Pentland, Recker, et al. (2020)
The MinAdept clustering approach for discovering reference process models out of process variants	C. Li et al. (2010)
Workflow clustering using semantic similarity measures	Bergmann et al. (2013)
Discovering workflow changes with time-based trace clustering	Accorsi & Stocker (2011)
Detecting Change in Processes Using Comparative Trace Clustering.	B. F. A. Hompes et al. (2015a)
Process mining through artificial neural networks and support vector machines	Maita et al. (2015)
A two-step clustering approach for improving educational process model discovery	Ariouat et al. (2016)
Clustering event traces by behavioral similarity	Koschmider (2017)
Similarity-based approaches for determining the number of trace clusters in process discovery	Koninck & Weerdt (2017)
Change Detection in Event Logs by Clustering	Koschmider & Moreira (2018)
Anomaly detection based on control-flow pattern of parallel business processes	Darmawan et al. (2018)
act2vec, trace2vec, log2vec, and model2vec: Representation Learning for Business Processes	De Koninck et al. (2018)
k-process: Model-Conformance-based Clustering of Process Instances.	Richter, Wahl, et al. (2019)
Visualizing Business Process Evolution	Yeshchenko et al. (2020)
Model-Aware Clustering of Non-conforming Traces	Richter, Zellner, et al. (2019)
Multi-Perspective Clustering of Process Execution Traces	Jablonski et al. (2019)
A non-compensatory approach for trace clustering	Delias et al. (2019)

B Appendix B

Figure B: Comparing AMPI and BI for BPI 2020



Note: Compared to the BI, the ratio of inefficient cases identified by AMPI differs. BI also detects the overall difference between the two types of trips, but it labels 13% of domestic trips and only 30% of international trips as inefficient. The difference in the international travels can be explained by the fact that BI only evaluates the presence of inefficient activities, but not the location of non-inefficient activities. As outlined above, many cases do not start with submitting a permission for international travels, which does not correspond to the optimal flow of activities, but is not identified as inefficient behavior by BI.

C Appendix C

C.1 Additional Information on Alpha

Alpha has 15 plants across six countries. Nine plants are located in Germany, followed by two plants in the Netherlands, and one each in Austria, Italy, Poland, and the United Kingdom. Figure 8 shows a production terminal with a barcode scanner located on one of Alpha's production floors. Such terminals are operated by frontline employees and the scanner data are directly fed into Alpha's centralized IT system. These granular data represent the basis of our empirical analysis. In its aggregated form the data are shown on information screens throughout the plants as shown in Figure 9.

Figure 8: Production Terminal with Barcode Scanner

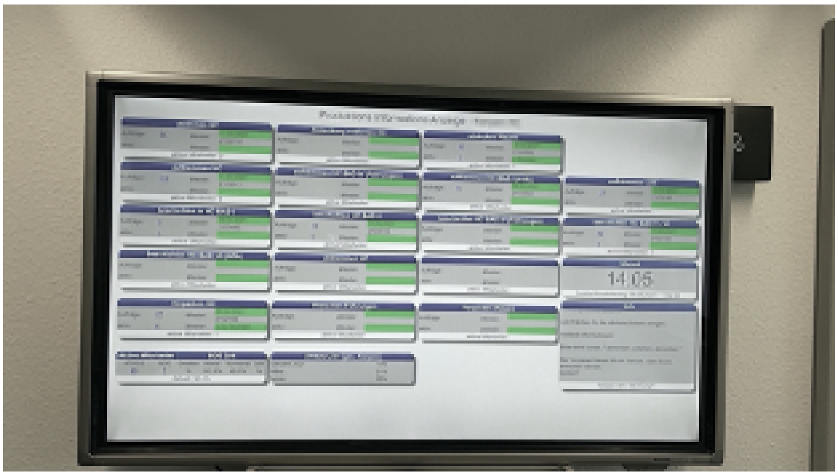


Figure 9: Information Screen Showing the Number of Completed, Due, and Overdue Orders

D Appendix D

D.1 Robustness Checks

Table 23: Additional Lags and Leads for Any Error

Dependent Variables	Reporting accuracy		log(Throughput time)	
Model	(1) ±7 Days OLS	(2) ±14 Days OLS	(3) ±7 Days OLS	(4) ±14 Days OLS
Any Error$_{t+14}$		0.001 (0.001)		0.025 (0.021)
Any Error$_{t+13}$		0.001 (0.001)		0.016 (0.017)
Any Error$_{t+12}$		-0.001 (0.001)		0.039+ (0.021)
Any Error$_{t+11}$		0.000 (0.001)		0.008 (0.018)
Any Error$_{t+10}$		-0.000 (0.001)		0.024 (0.018)
Any Error$_{t+9}$		0.001 (0.001)		0.053** (0.021)
Any Error$_{t+8}$		-0.001 (0.001)		0.025 (0.019)
Any Error$_{t+7}$	-0.002 (0.001)	-0.002 (0.001)	0.038* (0.018)	0.025 (0.018)
Any Error$_{t+6}$	-0.001 (0.001)	-0.001 (0.001)	0.070** (0.022)	0.059** (0.021)
Any Error$_{t+5}$	-0.000 (0.001)	-0.000 (0.001)	0.100*** (0.023)	0.090*** (0.021)
Any Error$_{t+4}$	0.001 (0.001)	0.001 (0.001)	0.124*** (0.021)	0.113*** (0.021)
Any Error$_{t+3}$	-0.002 (0.001)	-0.002 (0.001)	0.105*** (0.018)	0.095*** (0.017)
Any Error$_{t+2}$	-0.000 (0.001)	-0.000 (0.001)	0.075*** (0.019)	0.067*** (0.019)
Any Error$_{t+1}$	-0.002+ (0.001)	-0.002+ (0.001)	-0.029 (0.020)	-0.035+ (0.019)
Any Error$_t$	**0.002** **(0.001)**	**0.002** **(0.001)**	**-0.033** **(0.021)**	**-0.038+** **(0.021)**
Any Error$_{t-1}$	0.003** (0.001)	0.003** (0.001)	-0.105*** (0.021)	-0.112*** (0.021)
Any Error$_{t-2}$	0.003* (0.001)	0.003* (0.001)	-0.061** (0.021)	-0.071*** (0.020)
Any Error$_{t-3}$	0.003** (0.001)	0.003** (0.001)	0.001 (0.020)	-0.008 (0.019)
Any Error$_{t-4}$	0.002* (0.001)	0.002+ (0.001)	-0.018 (0.018)	-0.028 (0.019)
Any Error$_{t-5}$	0.001 (0.001)	0.001 (0.001)	0.021 (0.020)	0.01 (0.020)
Any Error$_{t-6}$	0.000 (0.001)	0.000 (0.001)	0.029 (0.019)	0.017 (0.019)
Any Error$_{t-7}$	-0.000 (0.001)	-0.000 (0.001)	0.032 (0.022)	0.017 (0.021)
Any Error$_{t-8}$		0.001		-0.009

		(0.002)		(0.020)
Any Error$_{t-9}$		0.000		0.005
		(0.001)		(0.021)
Any Error$_{t-10}$		-0.002+		0.019
		(0.001)		(0.019)
Any Error$_{t-11}$		0.000		-0.002
		(0.002)		(0.019)
Any Error$_{t-12}$		-0.001		0.047*
		(0.001)		(0.019)
Any Error$_{t-13}$		-0.000		-0.003
		(0.002)		(0.018)
Any Error$_{t-14}$		0.000		0.029
		(0.001)		(0.021)
Reporting Accuracy$_{[Prev.-Week Avg.]}$	0.786***	0.786***		
	(0.018)	(0.018)		
log(Throughput Time)$_{[Prev.-Week Avg.]}$			0.691***	0.691***
			(0.012)	(0.012)
Time Fixed Effects	Yes	Yes	Yes	Yes
Plant Fixed Effects	Yes	Yes	Yes	Yes
Number of Observations	141,500	141,496	141,500	141,496
Number of Articles	5,965	5,965	5,965	5,965
(Adj.) R-squared	0.668	0.668	0.524	0.524

Note:
The table presents IV regressions with weighted number of errors for other articles by the same customer used as an instrument for the reporting of any error. The sample used in col. 1 is derived from the sample in cols. 2-5. All regressions include year, month, and day-of-the-week fixed effects. Standard errors clustered by article are reported in parentheses. *** $p < 0.001$, + $p < 0.10$.

Table 24: Lags and Leads for Number of Errors

Dependent Variables	Reporting accuracy	log(Throughput time)
Model	(1) ±7 Days OLS	(3) ±7 Days OLS
log(1+Number of Errors)$_{t+7}$	-0.003 (0.002)	0.039+ (0.022)
log(1+Number of Errors)$_{t+6}$	0.000 (0.001)	0.083** (0.026)
log(1+Number of Errors)$_{t+5}$	-0.001 (0.001)	0.107*** (0.026)
log(1+Number of Errors)$_{t+4}$	0.000 (0.002)	0.135*** (0.023)
log(1+Number of Errors)$_{t+3}$	-0.003 (0.002)	0.122*** (0.021)
log(1+Number of Errors)$_{t+2}$	-0.001 (0.001)	0.076** (0.025)
log(1+Number of Errors)$_{t+1}$	-0.002 (0.002)	-0.024 (0.025)
log(1+Number of Errors)$_{t}$	**0.002** **(0.002)**	**-0.031** **(0.024)**
log(1+Number of Errors)$_{t-1}$	0.004* (0.002)	-0.130*** (0.028)
log(1+Number of Errors)$_{t-2}$	0.002+ (0.001)	-0.076** (0.027)
log(1+Number of Errors)$_{t-3}$	0.004** (0.001)	0.009 (0.026)
log(1+Number of Errors)$_{t-4}$	0.002 (0.002)	-0.019 (0.022)
log(1+Number of Errors)$_{t-5}$	0.002 (0.002)	0.035 (0.025)
log(1+Number of Errors)$_{t-6}$	0.001 (0.002)	0.039 (0.024)
log(1+Number of Errors)$_{t-7}$	-0.000 (0.001)	0.048+ (0.028)
Reporting Accuracy$_{\text{[Prev.-Week Avg.]}}$	0.786*** (0.018)	
log(Throughput Time)$_{\text{[Prev.-Week Avg.]}}$		0.691*** (0.012)
Time Fixed Effects	Yes	Yes
Plant Fixed Effects	Yes	Yes
Number of Observations	141,500	141,500
Number of Articles	5,965	5,965
(Adj.) R-squared	0.668	0.524

Note:
Note: OLS regressions are presented in cols. 2,3. All regressions include year, month, and day-of-the-week fixed effects. Standard errors clustered by article reported in parentheses. *** p < 0.001, ** p < 0.01, * p < 0.05, + p < 0.10.

Table 25: All Steps Reported

Model	All Steps Reported (1=yes, 0=no) (1) LPM	(2) Logit	(3) LPM	(4) Logit	(5) LPM	(6) Logit
	LPM	Logit	LPM	Logit	LPM	Logit
Error_{t-1}	0.014*	0.352*	0.011*	0.103		
	(0.005)	(0.139)	(0.005)	(0.163)		
First Error_{t-1}					0.028***	0.760*
					(0.008)	(0.383)
Reporting $\text{Accuracy}_{[Prev.-WeekAvg.]}$			2.630***	20.147***	2.630***	20.153***
			(0.083)	(0.928)	(0.083)	(0.928)
Time Fixed Effects	Yes	Yes	Yes	Yes	Yes	Yes
Plant Fixed Effects	Yes	Yes	Yes	Yes	Yes	Yes
Number of Observations	265,098	264,740	141,500	141,366	141,500	141,366
Number of Articles	7,818	7,806	5,965	5,955	5,965	5,955
(Pseudo) Adj. R-squared	0.267	0.374	0.664	0.660	0.664	0.661

Note:
LPM estimates are presented in cols. 1, 3, and 5. Logit estimates are presented in cols. 2, 4, and 6. All regressions include year, month, and day-of-the-week fixed effects. Standard errors clustered by article are reported in parentheses. The number of observations differs because of the inclusion of previous week average in reporting accuracy in cols. 3-6. The difference in sample size between LPM and logit emerges because observations without variation in the outcome are dropped for logit regressions. *** $p < 0.001$, * $p < 0.05$.

Table 26: Additional Control Variables

Dependent Variables	Reporting accuracy			log(Throughput time)		
Model	(1) OLS	(2) OLS	(3) OLS	(4) OLS	(5) OLS	(6) OLS
Error_{t-1}	0.004**	0.004**	0.004***	-0.075**	-0.066**	-0.081***
	(0.001)	(0.001)	(0.001)	(0.024)	(0.024)	(0.024)
Time Trend	0.000***			0.000+		
	(0.000)			(0.000)		
log(1 + Number of Cases)		-0.001*			0.013	
		(0.000)			-0.016	
Reporting $\text{Accuracy}_{[Prev.-WeekAvg.]}$	0.789***	0.786***				
	(0.018)	(0.018)				
Reporting $\text{Accuracy}_{[Prev.-2WeekAvg.]}$			0.798***			
			(0.017)			
$\text{log(Throughput Time)}_{[Prev.-WeekAvg.]}$				0.687***	0.692***	
				(0.012)	(0.012)	
$\text{log(Throughput Time)}_{[Prev.-2WeekAvg.]}$						0.690***
						(0.011)
Time Fixed Effects	No	Yes	Yes	No	Yes	Yes
Plant Fixed Effects	Yes	Yes	Yes	Yes	Yes	Yes
Number of Observations	141,500	141,500	162,003	141,500	141,500	162,003
Number of Articles	5,965	5,965	6,602	5,965	5,965	6,602
(Pseudo) Adj. R-squared	0.668	0.668	0.674	0.523	0.478	0.514

Note:
OLS regressions are presented in cols. 1-6. All regressions include year, month, and day-of-the-week fixed effects. Standard errors clustered by article reported in parentheses.*** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$, + $p < 0.10$.

Table 27: Variating the Minimum Number of Batches

Dependent Variables	Reporting accuracy				log(Throughput time)			
Model	(1) >=20 OLS	(2) >=30 OLS	(3) >=40 OLS	(4) >=50 OLS	(5) >=20 OLS	(6) >=30 OLS	(7) >=40 OLS	(8) >=50 OLS
Any Error$_t$	0.001	-0.000	-0.000	-0.000	0.011	0.009	0.014	0.015
	(0.001)	(0.001)	(0.001)	(0.001)	(0.023)	(0.023)	(0.023)	(0.023)
Any Error$_{t-1}$	0.003**	0.004**	0.004**	0.003**	-0.039	-0.022	-0.018	-0.024
	(0.001)	(0.001)	(0.001)	(0.001)	(0.024)	(0.024)	(0.024)	(0.025)
Reporting Accuracy$_{[Prev.-WeekAvg.]}$	0.817***	0.832***	0.838***	0.847***				
	(0.017)	(0.017)	(0.017)	(0.016)				
log(Throughput Time)$_{[Prev.-WeekAvg.]}$					0.702***	0.707***	0.709***	0.711***
					(0.012)	(0.013)	(0.013)	(0.013)
Time Fixed Effects	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Plant Fixed Effects	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Number of Observations	136,357	132,255	128,846	125,700	136,357	132,255	128,846	125,700
Number of Articles	3,647	2,576	2,002	1,654	3,647	2,576	2,002	1,654
Adj. R-squared	0.692	0.704	0.712	0.719	0.536	0.542	0.545	0.547

Note:
OLS regressions are presented in cols. 1-8. All regressions include year, month, and day-of-the-week fixed effects. Standard errors clustered by article reported in parentheses. The header indicates the minimum number of times an article is produced to be included in the analysis (i.e., 20, 30, 40, and 50). *** p < 0.001, ** p < 0.01.

Table 28: Excluding September 2016 and the First Half of 2020

Dependent Variables	Reporting accuracy		log(Throughput time)	
Model	(1) Excl. Sep. 2016 OLS	(2) Excl. first half 2020 OLS	(3) Excl. Sep. 2016 OLS	(4) Excl. first half 2020 OLS
Any Error$_t$	0.002+	0.002	0.002	0.007
	(0.001)	(0.001)	(0.022)	(0.023)
Any Error$_{t-1}$	0.004**	0.003**	-0.073**	-0.062*
	(0.001)	(0.001)	(0.025)	(0.025)
Reporting Accuracy$_{[Prev.-WeekAvg.]}$	0.781***	0.799***		
	(0.019)	(0.018)		
log(Throughput Time)$_{[Prev.-WeekAvg.]}$			0.692***	0.695***
			(0.012)	(0.012)
Time Fixed Effects	Yes	Yes	Yes	Yes
Plant Fixed Effects	Yes	Yes	Yes	Yes
Number of Observations	139,849	130,016	139,849	130,016
Number of Articles	5,929	5,800	5,929	5,800
(Adj.) R-squared	0.659	0.688	0.523	0.528

Note:
OLS regressions are presented in cols. 1-4. All regressions include year, month, and day-of-the-week fixed effects. Standard errors clustered by article reported in parentheses. *** p < 0.001, ** p < 0.01, * p < 0.05, + p < 0.10.

Table 29: Errors and Reporting Accuracy Including Article Fixed Effects

	Main Effect (H1)		Interaction (H3a)		Heterogeneity	
Model	(1) OLS	(2) 2SLS	(3) OLS	(4) OLS	(5) OLS	(6) OLS
Any Error$_{t-1}$	0.002	0.016*	0			
	(0.001)	(0.007)	(0.001)			
Low$_{t-1}$			-0.002			
			(0.001)			
Any Error$_{t-1}$ ×Low$_{t-1}$			0.008**			
			(0.002)			
First Error$_{t-1}$				0.007**		
				(0.002)		
Error Corrective Action$_t$					0.003+	
					(0.002)	
Error Preventive Action$_t$						0.007**
						(0.003)
Reporting Accuracy$_{[Prev.-Week Avg.]}$	0.695***	0.666***	0.695***	0.695***	0.695***	0.695***
	(0.028)	(0.037)	(0.028)	(0.028)	(0.028)	(0.028)
Time Fixed Effects	Yes	Yes	Yes	Yes	Yes	Yes
Article Fixed Effects	Yes	Yes	Yes	Yes	Yes	Yes
Number of Observations	141,500	91,555	141,500	141,500	141,500	141,500
Number of Articles	5,965	4,090	5,965	5,965	5,965	5,965
(Adj.) R-squared	0.399	0.338	0.399	0.399	0.399	0.399

Note:
All regressions include year, month, and day-of-the-week fixed effects. Standard errors clustered by article reported in parentheses. Col. 2 shows the results of an IV regression for the years 2017-2021. *** p < 0.001, ** p < 0.01, * p < 0.05, + p < 0.10.

Table 30: Errors and Throughput Time Including Article Fixed Effects

	Main Effect (H2)		Interaction (H3b)		Heterogeneity	
Model	(1) OLS	(2) 2SLS	(3) OLS	(4) OLS	(5) OLS	(6) OLS
Any Error$_{t-1}$	-0.070***	-0.077	-0.077***			
	(0.020)	(0.108)	(0.022)			
Low$_{t-1}$			0.005			
			(0.014)			
Any Error$_{t-1}$ $\times$Low$_{t-1}$			0.037			
			(0.048)			
First Error$_{t-1}$				-0.067		
				(0.041)		
Error Corrective Action$_t$					-0.123***	
					(0.029)	
Error Preventive Action$_t$						-0.096+
						(0.049)
log(Thtoughput Time)$_{[Prev.-WeekAvg.]}$	0.374***	0.373***	0.374***	0.374***	0.374***	0.374***
	(0.011)	(0.014)	(0.011)	(0.011)	(0.011)	(0.011)
Time Fixed Effects	Yes	Yes	Yes	Yes	Yes	Yes
Article Fixed Effects	Yes	Yes	Yes	Yes	Yes	Yes
Number of Observations	141,500	91,555	141,500	141,500	141,500	141,500
Number of Articles	5,965	4,090	5,965	5,965	5,965	5,965
(Adj.) R-squared	0.173	0.174	0.173	0.173	0.173	0.173

Note:
All regressions include year, month, and day-of-the-week fixed effects. Standard errors clustered by article reported in parentheses. Col. 2 shows the results of an IV regression for the years 2017-2021. *** p < 0.001, + p < 0.10.

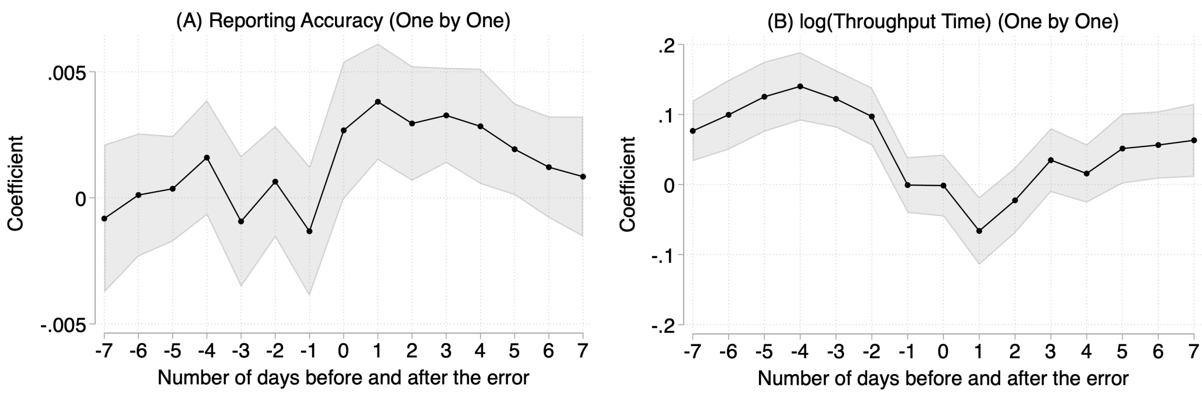


Figure 10: Errors and Routine Performance, Lags and Leads of Errors Included One by One.

E Appendix E

E.1 Additional Information on Instrumental Variable Approach

Table 31: Errors Reported by Customer 75259 on 29 March 2017

Error No.	Description	Plant	Surface Treatment	Article Id	Date	Order Id	Customer Id
116	Wrong spraying technique	10	Painting	320071	2017-03-29	672875	75259
86	Wrong selection of mixed batch	10	Tufram®coating	132405	2017-03-29	671156	75259
49	Wrong production documents	10	Staining	189857	2017-03-29	674740	75259

F Appendix F

F.1 Difference-in-Differences Approach

To capture responses to errors using a difference-in-differences approach, we consider changes in the dependent variables between days before $(t-1)$ and after $(t+1)$, an error on day t. Following Foerderer & Schuetz (2022), we kept only an article's first error in the sample to avoid multiple treatments of the same article. To construct clean pre- and post-treatment windows, we only consider panel observations with the first error on day t and no other errors on days $t-2$, $t-1$, $t+1$, and $t+2$. We set this restriction to avoid other errors directly preceding or succeeding our measurements, influencing the results. Control groups, in contrast, have no prior errors up to $t+2$. To avoid overlapping observations, we construct control groups so that the control groups of the same article do not use the same observation twice, i.e., for $t-1$ and $t+1$. Where $t-1$ and $t+1$ overlap, we only keep the first matching group in chronological order.

The above process results in 372 treatment groups and 4,681 control groups that match the selection criteria. To achieve a balanced sample, we use one-on-one nearest-neighbor matching in the Mahalanobis distance (which is a well-established distance measure that has been widely used for matching procedures in management research; Cui et al. (2020), Friberg & Sanctuary (2017), Liu et al. (2019)) on average daily throughput time (log-transformed), and average daily reporting accuracy up to and including each matching point on day $t-1$. We use the MatchIt package in R (version 4.2.0) for the matching procedure (Stuart et al., 2011). After this procedure, we have 369 matching treatment and control groups with two observations per group ($t-1$ and $t+1$). The results in Table 32 show that the approach was effective at removing the imbalance in observable features. In Table 33, we report the results of the difference-in-differences analysis. The results are consistent with our main analysis and they indicate that the treatment group has higher reporting accuracy and lower throughput time after an error has been reported.

Table 32: Characteristics of Treatment and Control Groups Before and After Matching

	(1) Before Matching			(2) After Matching		
Measure	Treatment	Control	t-statistic	Treatment	Control	t-statistic
Avg. Daily Reporting Accuracy	0.973	0.954	-3.554***	0.972	0.979	1.178
Avg. Daily log(Throughput Time)	3.434	3.500	1.352	3.434	3.433	-0.014
Number of Observations	372	4,681		369	369	

Note:
Matching based on one-on-one nearest neighbor matching in the Mahalanobis distance. *** $p < 0.001$.

Table 33: Difference-in-Differences Results

	Reporting Accuracy			log(Throughput Time)		
Model	(1) Full OLS	(2) Matched OLS	(3) Matched + lagged DV OLS	(4) Full OLS	(5) Matched OLS	(6) Matched + lagged DV OLS
Treat	0.011+	-0.003	-0.001	0.125+	0.257***	0.188**
	(0.006)	(0.006)	(0.004)	(0.065)	(0.077)	(0.071)
After	-0.001	-0.004*	-0.003	-0.079***	-0.020	0.038
	(0.001)	(0.002)	(0.002)	(0.015)	(0.043)	(0.053)
Treat $\times$After	0.013**	0.015***	0.013**	-0.185**	-0.235**	-0.355***
	(0.004)	(0.004)	(0.005)	(0.060)	(0.073)	(0.089)
Reporting Accuracy$_{[Prev.-WeekAvg.]}$			0.613***			
			(0.087)			
log(Throughput Time)$_{[Prev.-WeekAvg.]}$						0.684***
						(0.037)
Time Fixed Effects	Yes	Yes	Yes	Yes	Yes	Yes
Plant Fixed Effects	Yes	Yes	Yes	Yes	Yes	Yes
Number of Observations	10,106	1,476	1,233	10,106	1,476	1,233
Number of Articles	1,282	528	528	1,282	528	528
Adj. R-squared	0.472	0.316	0.639	0.179	0.185	0.478

Note:
OLS regressions are presented in cols. 1-6. All regressions include year, month, and day-of-the-week fixed effects. Standard errors clustered by article are reported in parentheses. The number of observations differs because of the implementation of one-on-one matching and the additional inclusion of previous week averages of the dependent variables. *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$, + $p < 0.10$

CURRICULUM VITAE

02/2018-12/2023	Ph.D. in Business Administration	University of Mannheim, Mannheim
02/2018-09/2022	Research assistant at the chair of Enterprise Systems	University of Mannheim, Mannheim
03/2014-04/2016	Master of Science in Business Informatics	University of Mannheim, Mannheim
09/2010-02/2012	Master of Science in Electronic Commerce	Shiraz University, Shiraz
09/2006-02/2010	Bachelor of Science in Information Technology Engineering	Azad University, Sanandaj