博士論文

Framework for data-driven process improvement and operations support in biopharmaceutical drug product manufacturing

（バイオ医薬品製剤製造におけるデータ駆動型プロセス改善・運転支援のためのフレームワーク）

カゾーラ　ジオエレ

Framework for data-driven process improvement and operations support in biopharmaceutical drug product manufacturing

Thesis written by Gioele CASOLA

Supervisor:

Prof. Dr. Hirokazu SUGIYAMA

Co-examiners:

Prof. Dr. Masahiko HIRAO

Dr. Dimitrios GEROGIORGIS

Prof. Dr. Taichi ITO

Prof. Dr. Shigeo Ted OYAMA

Prof. Dr. Yasuyuki SAKAI

Thesis submitted for the degree of Doctor of Philosophy

Department of Chemical System Engineering

Graduate School of Engineering

The University of Tokyo

2018

Abstract

Globally, in developed countries, the aging population carries several consequences, such as manufacturing, services, taxes, and healthcare. In parallel, the cost of R&D persistently levitates, whereas the cost of manufacturing remains constant; therefore, large pharmaceutical companies have to face the increasing competition of generic producers. To contrast the resulting decrease of sales margin, big pharmaceutical companies are required to improve the performance of their processes, e.g., by achieving reduced production time, and down time as well as enhanced capacity.

Various Process Systems Engineering (PSE) approaches, which comprise modeling simulation and optimization have been applied to achieve the optimal design in the synthesis of pharmaceutical processes. A further concept, namely digitalization or Industry 4.0 (I4.0), is the current major driver of the improvement of processes and the technological revolution in the industry. More rigorous approaches and introspective studies on the incorporation of well-established PSE methodologies with the novel digital revolution approaches are necessary to unlock the real potential of I4.0 in improving pharmaceutical manufacturing processes.

The thesis presents a framework, the application of which assists the uncertainty-conscious and data-driven decision-making in the process improvement and operations support in Good Manufacturing Practice (GMP)-controlled biopharmaceutical manufacturing. The framework consists of three main parts: data preprocessing (I), process performance assessment (II), and predictive maintenance decision-making (III). The framework was applied in an industrial case study, where manufacturing records generated from a change over-process operated in a facility belonging to F. Hoffmann-La Roche Ltd.

In biopharmaceutical Drug Product (DP) manufacturing, sterile filling plants are usually non-dedicated, which implies the necessity of change-over operations. The sterile filling process comprises washing and sterilizing of empty glass containers, filling of these containers with a drug solution, sealing of the containers, and finally the visual inspection. Start-up and changer-over are support processes, which enable the switch from a product/format to another by maintaining the sterility and purity required for the manufacturing of high-quality products. An example of a support process is the Cleaning-In-Place and Sterilizing-In-Place (CIP/SIP), which involves cleaning the product-contacting surface of the filling system such as pipes, tanks, and filling needles

by removing product residues and particles, sterilizing it, and finally drying it. Support processes are extremely time intensive, approximately half of the production timeline; in fact is the underlying reason of this case study.

In the first part of the framework, a new algorithm for transforming raw data recorded in biopharmaceutical manufacturing into functional data in an automated manner is presented. The algorithm consists of seven activities, including process task identification by natural language recognition, model training for selecting and filtering of noise—e.g., data not belonging to commercial batches—from the raw data, and clustering the process into single batches using semi-supervised clustering. The remaining activities ensure the compatibility of the resulting dataset with the remaining part of the framework.

The raw data string was treated similar to a DNA strain during sequencing: first, the process recipe was randomly cut into primers, which are sequences of strings and fragments of the raw data string; the heterogeneity coefficients between the primers were calculated. The heterogeneity coefficient and the primer size were then used to train decision tree-classifiers based on the human perception of data noise. The training of the classification model resulted in the identification of the noise in the data with an F-score of 0.99. The filtering of the noise could shrink the data set to 60% of the original size; 40% of the data recorded did not contain process relevant information but was recorded because of GMP regulations and a non-cost-efficient monitoring strategy. The noise-free data were clustered using a constraint k-means algorithm, which allowed separation of every single batch. Last, the data points were classified into three categories, namely, *normal process*, *failure*, and *repetition*. The resulting dataset was used in the subsequent two parts of the framework to identify the improvement potentials and to predict imminent failures.

In the second part of the framework, an uncertainty-conscious methodology was presented; it can assess process performance and facilitate process improvement in biopharmaceutical DP manufacturing. The work was described as a six-activity model using IDEF0, which are “define key performance indicators (KPIs) (A1),” “create an initial process performance model (A2),” “collect and adapt data (A3),” “characterize the process performance model (A4),” “identify tasks to improve (A5),” and “perform what-if analysis (A6)”.

The KPI was defined as the process runtime (A1) and was modeled as the sum of the duration of the process tasks, remedying operations (A2). The historical records were imported (A3 and A4). A two-loops stochastic global sensitivity analysis (GSA) employing a partial rank correlation coefficient (PRCC) was used to select the

tasks that highly affect the KPI. The GSA aimed to assess the process design in an operational environment, so the task duration was defined as the changeable parameter. To incorporate operational uncertainty in the analysis, the outer loop propagated the operation uncertainty by random sampling Monte Carlo simulation (MCS). In the inner loop, the PRCC for every task was calculated with Latin hypercube sampling-MCS (LHS-MCS) using the performance model. The feasibility indicator was defined as 0 to 1, 0 being complete infeasibility and 1 being perfect feasibility; the feasibility process modification reflects industrial know-how. Finally, the results of PRCC and the feasibility analysis were combined (A5) to identify tasks that showed high impact and feasibility as improvement potentials. The suggestions of industry experts were implemented in a what-if analysis with the process model (A6); the result showed 120 h potential time-savings.

In the third part of the algorithm, an intelligent algorithm, which is based on Industry 4.0 and big data approaches was presented. The algorithm predicts the failure status of the process from physical sensors in real-time. A decision tree (DT) model was trained to identify failed batches from successful ones; historical sensor data were transformed by principal component analysis (PCA) to reduce the dimensionality of the system. A retraining loop maintained the quality of the prediction over time because the algorithm is based on machine learning and the process continuously evolves; the algorithm resulted in a decision after the analysis of the risk on the performance in case of action. An integrated failure prediction and decision-making tool was used to support the decision of stopping a batch before a failure occurs. The deployment of the algorithm on the real-world data results in the potential time saving of approximately 100 hours per month.

The thesis proposes a data-driven framework for supporting the decision-making in process and operation improvement for biopharmaceutical manufacturing. The framework consists of three main steps: the integration of existing industrial databases, assessment of the process performance and identification of the task to improve, and imminent failure mitigation by plant predictive maintenance. The thesis aims to provide a novel and rigorous tool/approach that is applicable in an industrial environment, to solve long-term challenges, such as decision-making regarding maintenance policies and process improvement as well as daily operation challenges, such as downtime reduction. The result of the application of the framework in the industrial case study was implemented in the commercial facility; a reduction of the process runtime was achieved. In future works, the framework will be integrated in the manufacturing operations after further generalization. Moreover, the framework will be

expanded to other pharmaceutical processes, such as sterile filling, packaging, transportation. Last, the impact on sustainability will be included in the decision-making of process modification.

Acknowledgment

This thesis is the result of three years of hard work and intense traveling, which made me grow as a researcher, project manager, speaker, and more importantly as a person. This work, as well as my personal development, would not have been possible without the contribution of people who questioned, supported, and stimulated the creation of new ideas.

My most sincere gratitude goes to my supervisor, Hirokazu Sugiyama (Hiro). He gave me the opportunity to first travel to Japan as a master student and then offered me the great honor to be his first Ph.D. student. Hiro helped me develop my technical and communication skills by giving me the chance to attend various conferences and industrial meetings. This work would not have been possible without his constant support.

I am extremely grateful to Markus Mattern and Christian Siegmund from F. Hoffmann-La Roche Ltd. for believing in my ideas and supervising my work during the time I spent in the production site of Roche in Kasieraugst, Switzerland. Their invaluable feedbacks enabled me to develop my communication skills in a non-academic environment.

My special thanks go to Konrad Hungebühler of ETH in Zurich and Stavros Papadokostantakis of Chalmers University in Göteborg for introducing me to Hiro and supporting the international exchange during my master thesis project in Tokyo. They also contributed to the start-up of my Ph.D. project with very fruitful discussions and wise comments.

My deepest appreciation goes to Rainer Schmidt for always sustaining my project and for initiating the collaboration with F. Hoffmann-La Roche Ltd, without which the thesis would not have been possible.

My sincere thanks go to Masahiko Hirao, Dimitrios Gerogiorgis, Taichi Ito, Shigeo Ted Oyama, and Yasuyuki Sakai for accepting to review my thesis and providing their insightful comments.

Sincere thanks go to Andrea Mieskes, Tom Weigert, Stephanie Knueppel, Aldo Buschemi, Jens Kuehne, Marc Borer, Patrick Sieweck, Timur Guener, Maik Haering, Alexander Svensson, Frauke Schenuit, Sabrina Steiner,

Valentina Masi, Jeremy Struchen, Carmon Hesse, and many other employees of F. Hoffmann-La Roche Ltd. for their technical support and for facilitating my visits in the company.

I am profoundly grateful to the members of the Hirao-Sugiyama Laboratory for giving me continuous support and motivation and bearing my small, but continuous complains and comments. My appreciations go to Masahiko Hirao, Eri Amasawa, Sarah Badr, and all the attendants of the laboratory seminar where I could discuss my work and receive extremely perceptive feedbacks. Sincere thanks go to Akiko Kanatani who supported me with administration along these years. Sincere thanks go to Kensaku Matsunami, Keisho Yabuta, Harauku Shirahata, and Naoki Yokokawa per helping me during my days in Tokyo.

I would like to thank Anicia Zeberli who has been a great research fellow, neighbor, trip organizer, dinner companion, counselor, and, most importantly, a real friend during for last three years.

Finally, yet importantly, my deepest gratitude is reserved for my family and friends for the continuous support and for helping me to become the person, I am now. To my wife, Noemi, thank you for accepting my absence from home for such a long time and for always giving me your love, motivation, and support.

Tokyo, 26 July 2018 Gioele Casola

List of Contents

List of Figures

List of Tables

List of Acronyms

API	Active Pharmaceutical Ingredient
BADT	Bagged Decision Tree
CAPE	Computer-Aided Process Engineering
CFR	Code of Federal Regulation
cGMP	Current Good Manufacturing Practice
CIP	Cleaning-In-Place
DNA	Deoxyribonucleic Acid
DOO	Degree Of Overlap
DP	Drug Product
DS	Drug Substance
DT	Decision Tree
DW	Demineralized Water
ETS	Extremity Task Sequence
FDA	Food and Drug Administration
GMP	Good Manufacturing Practice
GSA	Global Sensitivity Analysis
ID	Identifier
IDEF0	Icam Definition for function modelling
IoT	Internet of Things
IT	Information Technology
KNN	K-Nearest Neighbor
KPI	Key Performance Indicator
LHS	Latin Hypercube Sampling
LSS	Lean Six Sigma

mAb	monoclonal Antibody
MCS	Monte Carlo Simulation
MES	Manufacturing Execution System
MSO	Modelling-Simulation-Optimization
NPV	Negative Predictive Value
PAT	Process Analytical Technology
PCA	Principal Component Analysis
PDF	Probability Distribution Function
PID	Proportional-Integral-Derivative
PN	Process-Noise
PRCC	Partial Rank Correlation Coefficient
PS	Pure Steam
PSE	Process System Engineering
QbD	Quality by Design
RABS	Restricted-Access Barrier System
RAM	Random-Access Memory
RCA	Root cause analysis
R&D	Research and Development
RNE	Risk of Non-Effect
ROC	Receiver Operating Characteristic
SIP	Sterilizing-In-Place
SKNN	Sub-space k-Nearest Neighbor
SVM	Support Vector Machine
UF	Unpredictable Failure
WF	Wagner-Fischer
WFI	Water For Injection

Glossary

Assembly-like processes	Processes that add value to the product at each task
Biopharmaceutical	An adjective used to describe pharmaceutical products that are produced by a biological/enzymatic process—i.e., fermentation.
Classifier / classification model	Machine learning model that divides the data points into discrete classes by their attributes.
Control boundaries	Specific physical quantities that control the process; if the sensor measures a quantity that crosses the boundary, the process stops.
Corrective maintenance	Maintenance scheduled after the occurrence of the failure; such a policy is used to "repair" the process/plant
Data-driven	Model/system/approach based on data and its interconnection where no phenomenal model is needed.
Decision boundary	The boundary delimiting the combination of predictors assigned to each class; whenever the boundary is crossed the decision/class changes.
Deployment dataset	Dataset used in the deployment of the model; this dataset set is used to test the applicability of the model under operation conditions
Dirty hold time	The validated time during which a sealed isolator is considered save to be operated. It determines the maximum duration of a campaign
Downtime	Time in which a machine is not available for use
Drug product	A medicine—i.e., API solution or tablet—before being packaged
Extremity task sequence	Sequence of task located at the beginning and end of the process recipe
GMP	Set of guidelines that control the manufacturing operation as well as the process design
Injectables/parenterals	Class of liquid drugs administered via an injection
Key performance indicator	Main indicator/quantity describing the performance of a process
Label	Information carrier; a label carries the prediction judgment on a data point
Latin hypercube sampling	Structured stochastic simulation sampling method
Lean six sigma tools	Set of tools applied in various industry sector to reduced process variability while reducing waste and unnecessary process complexity
Morphological root	Sequence of tasks that are in common to all process recipes

Nonroutine events	Unplanned events examples of which are manufacturing failures and process repetition
Operational uncertainty	Uncertainty that is attributed to manufacturing operations, such as human error or accident
Pharmacovigilance	Detection, monitoring, assessment, and control of drug effects during the final phase of human trials and drug commercialization
Preventive Maintenance	Maintenance scheduled by the estimated probability of failure of the process; such a policy schedules maintenance regularly to prevent the occurrence of failures.
Predictor	Feature of the data that allow a prediction; a classifier uses predictors to deliver predictions
Predictive Maintenance	Maintenance scheduled by a simulated prediction of a failure from a model that describes the status of the process; such a policy schedules maintenance only when it is necessary.
Process space	Data space in which all the process data are observed. It is wider than the space limited by the control boundaries
Purity	The status of the product in the absence of contaminants, such as other products, or side products.
QbD	Systematic approach for ensuring the product quality by the process and product design
Repetition matrix	Matrix defining what is repeated in case of process failure
Sensitivity	Capability of a model to differentiate true positive from false negative predictions
Small molecule	A molecule which has size inferior to 1000 Dalton
Specificity	Capability of a model to differentiate true negative from false positive predictions
Sterility	The status of a product in the absence of pathogens, such as bacteria, or viruses.
Supervised machine learning	Machine learning model that is trained on labeled data
Unsupervised machine learning	Machine learning model that is trained on unlabeled data
Wagner-Fischer distance	Quantitative distance between two strings/number of different letters between two strings

Chapter 1: Introduction

1.1 Pharmaceutical manufacturing

1.1.1 Socio-economic review

Globally in developed countries, life quality and life expectancy at birth have been continuously increasing in previous years[1]; on the contrary, the fertility rates are following the opposite trend.[2] The combination of these two factors is leading to a constantly aging population,[3] which is expected to peak in the early 2020s owing to the "baby boomers born between the 1940s and the 1960s. Goldstone (2006) reported that in 2050 the oldest countries worldwide, i.e., Japan, Germany, Italy, and Switzerland, will have on an average 36% and 14% over-60 and under-15 population, respectively.[4] This will result in only 50% of the population being productive, which is 10% less than the ratio for the population aged 15 to 59 years from 2005. Such a structural change has consequences on several aspects, such as in manufacturing, services, taxes, and healthcare; healthcare especially plays a crucial role in a society that is not willing to perish. Because of the aging society, the amount of medication and services required for maintaining or even increasing, life expectancy is increasing, which leads to the rise of costs. Boecking et al. (2012) predicted an increase of 26% of the expenses for prescription drugs in an analysis on the pharmacoeconomic impact on pharmaceutical expense caused by the demographic change in Germany and France.[5] The prediction on the increase of healthcare expenses was performed with the assumption that the expenses per capita per age will not change with time, which represents the best-case scenario.

A prediction of a worst-case scenario is challenging because it is not possible to predict the additional cost resulting from the development of new drugs in the next 30 years. However, considering that the Research and Development (R&D) expenses are continuously increasing,[6] it is not difficult to imagine that the cost of the healthcare system will behave similarly. A compendium of fact and figures from the International Federation of Pharmaceutical Manufacturers and Association reports that the R&D expenses constantly levitated and reached 120% in the years between 2005 and 2015.[6] One of the reasons for such an increase in costs is the number of new drugs approved in the same timeframe; 28 drugs in 2005 and 56 in 2015, an essential part of which targets age-associated diseases, such as cancer, diabetes, and cardiovascular diseases. In another review, the European Federation of Pharmaceutical Industries and Association (2017) reported that the pharmaceuticals

& biotechnology sector spent 15% of the net sales in R&D activities in 2015, which is significantly larger than the 3.8% spent by all industries together.[7]

In contrast to the trend of R&D costs, manufacturing cost remained constant at around 30% of the cost of goods sold from 1994 to 2005 as reported by Basu et al. (2008).[8] To some extent, pharmaceutical companies did not face disruptive changes in how they operate—i.e., pharmaceutics is a very stable industry sector[9]—or introduce operation innovation by changing technology in the last ten years; therefore it can be safely assumed that the relative cost of manufacturing did not drastically change. However, large pharmaceutical companies have to face the increasing competition of generic producers, which do not have R&D expenditures and can optimally design the process, to benefit and reinvest in new R&D. Generics and biosimilar products, which are supported by various governments with the intent to decrease the public expenditure on healthcare,[10,11] have increased their reach from 2000 with a maximum 70% of volume share in Germany and Romania.[12]

Near future therapeutic innovations, such as cell therapy[13] and personalized medicine[14], are already forcing the big players to change their approach toward drug manufacturing. New technologies, such as machinery for cell cultures, will enter the market and the variety of drugs will proliferate, because each patient will receive a tailored drug. To be successful in this new challenge, it is of primary importance to improve the manufacturing process by reducing downtime and unnecessary operations and increasing productivity. It is crucial to change the culture by welcoming the changes.

1.1.2 Pharmaceutical manufacturing processes

Pharmaceutical manufacturing processes are the result of a very long effort, which usually lasts 13 years,[7] starting from the patent application and ending with the design of the process. The first ten are invested in the product and the authorization for marketing it, following which three years are dedicated to administrative procedures and finally the design of the commercial process. The total duration of a patent is 25 years, and this means that companies have to return on their investment in 12 years; however, before that, the generics enter the market, leaving little time for optimal process design. Sub-optimal processes are improved continuously throughout the operations; teams of operators, mechanics, and analysts work together until the optimality of the process is reached.[15]

As shown in **Figure 1.1**, pharmaceutical processes are classified by their products, with their Active Pharmaceutical Ingredients (API) divided into two classes small molecules and biopharmaceuticals. Small molecule APIs are chemically produced compounds such as paracetamol, acetylsalicylic acid, and codeine, whereas biopharmaceutical APIs are compounds, such as monoclonal Antibodies (mAb), proteins and hormones, produced by bacterial fermentation (see **Figure 1.1**). In the field, the synthesis of an API is named Drug Substance (DS) manufacturing and it comprises fermentation and purification processes.[16] After the synthesis of the API, the DS is transformed into a Drug Products (DP), which is the drug administrable to the patient and usually has two dosage forms: liquid and solid. Small molecule DPs can be found in both liquid (e.g., syrups, and injectables) and solid (e.g., tablets, capsules, and powders) forms; by contrast, biopharmaceutical DPs are mostly found in liquid formulation. For conciseness and because of the lack of related research works, this study will only focus on biopharmaceutical DP manufacturing.

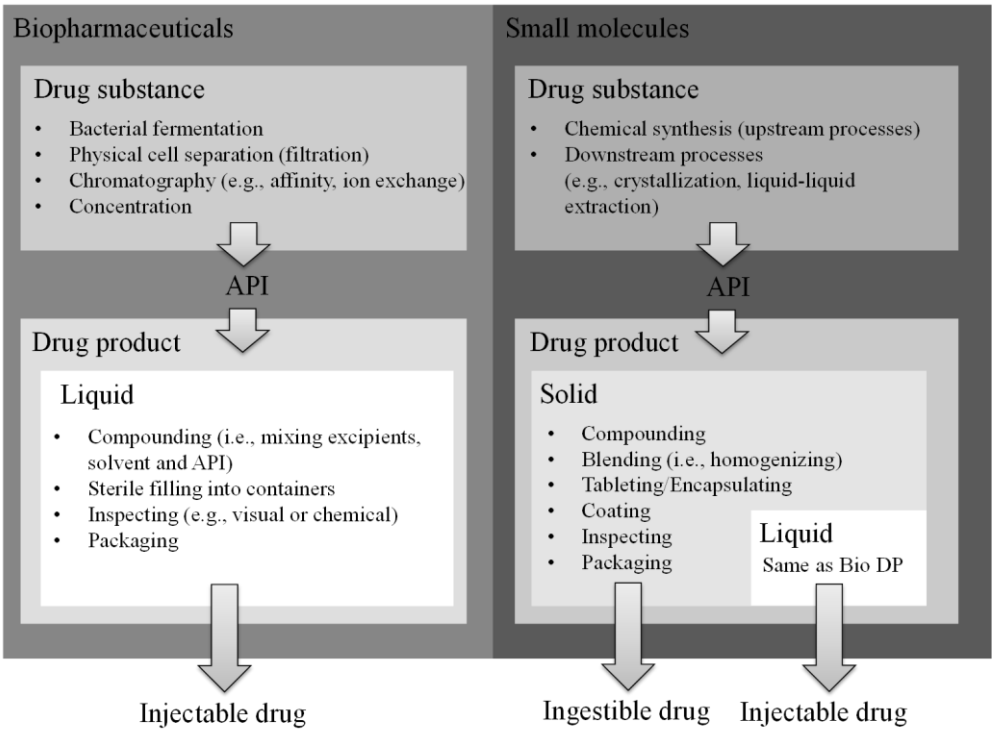


Figure 1.1 Classification of pharmaceutical products and processes.

In biopharmaceutical DP manufacturing a DS is put in solution and the mixture is filled in containers in the sterile environment, because the drug has to be directly injected into the patient's body. The filling process consists of washing and sterilizing the containers, e.g., glass vials or syringes, filling the containers with the DP solution, sealing the containers with plastic stoppers, capping, and visual inspection. The DP solution is continuously maintained in a sterile condition by aseptic technologies, such as isolators and Restricted-Access Barrier Systems (RABS) until the containers are sealed.[17] Filling facilities are usually non-dedicated, meaning that they can accommodate the production of multiple products with different container formats; this implies in addition to production scheduling, there is a necessity of plant flexibility—i.e., mounting and unmounting of apparatuses—and change-over operations.

The product solutions are filled batch-wise; filling batches that have the same or similar products (P_a, P_b, P_c, in **Figure 1.2**) are filled in a campaign, which is defined as the timeframe where the isolator remains sealed and in sterile conditions. The campaign starts when the isolator is decontaminated and is concluded when the isolator is opened; during a campaign, change-over operations are executed in the sterile environment through a glove-box system. The length of a campaign varies from one batch to a number of batches fillable within the dirty hold time, the validated maximum time during which the isolator is considered free of contaminant and operational.

Start-up and changer-over operations are support processes, manual or automated, which enable switching from a product/format to another while maintaining the conditions—i.e., sterility and purity—required for manufacturing high-quality products. **Figure 1.2** shows an explicative representation of a multi-product/multi-format filling campaign, with different colors representing different formats or products; the campaign starts with start-up operations and the filling batches are followed by change-over operations. Besides the manual mounting of the format-specific part, support operations comprise isolator decontamination and piping cleaning, sterilizing, and drying processes. The first process ensures that the isolator is free of pathogens by providing a vaporized H_2O_2 solution in the production environment;[18] the latter processes, which are referred to as Cleaning-In-Place and Sterilizing-In-Place (CIP/SIP), consist of cleaning the product-contacting surface of the filling system by removing product residues and particles, sterilizing it, and finally drying it.

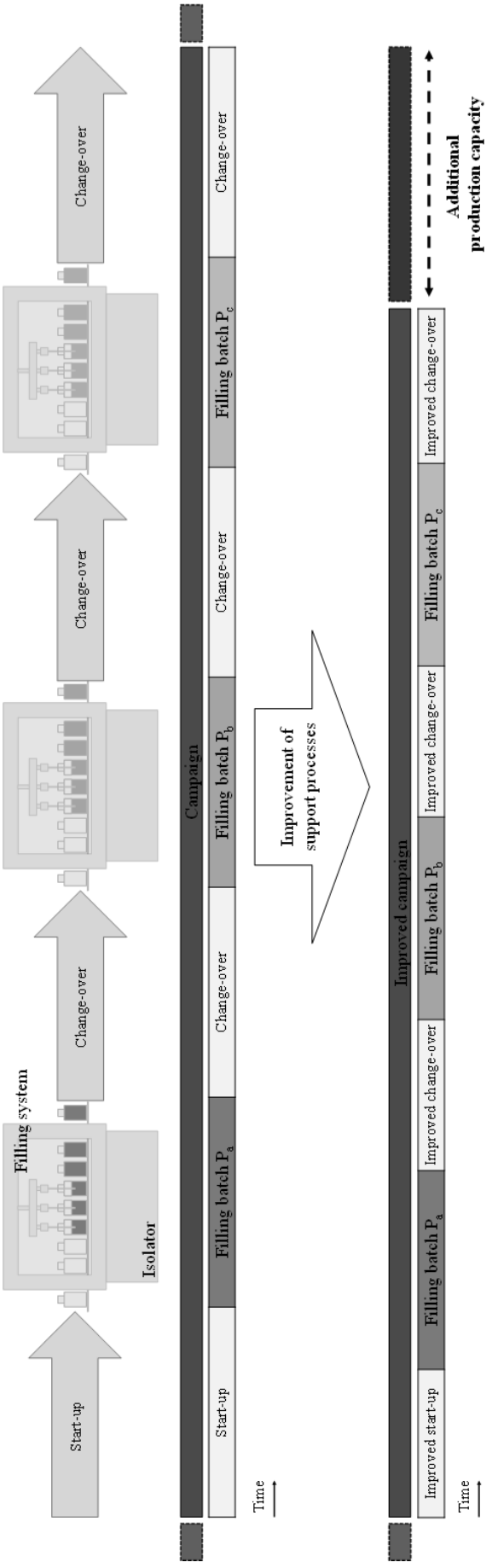


Figure 1.2 Representation of a multi-format/product filling campaign.

The terminology "-in-place" indicates for the fact that the filling system, which consists of pipes, buffer tanks, and filling needles, is cleaned and sterilized in the assembled status only shortly before the production is started—i.e., when the pipes are in place.

Support processes are very time intensive; in fact, it can be estimated that these processes require half of the production timeline leaving the remaining half the actual production. From experience, on average, depending on the size of the filling batch, filling and change-over operations both require around 8 hours. By assuming that the production only consists of these two operations, half of the production time is invested in supporting the manufacturing. Therefore, the improvement of the support operation performance, i.e., reduction of the runtime, could help increase the potential productivity or production capacity of the facility, as highlighted in **Figure 1.2**. The time-intensity can currently be attributed to the highly controlled environment, which is necessary to ensure that products do not cause harm to the patient; Good Manufacturing Practice (GMP) is a set of guidelines that ensures the quality of the products is achieved throughout the lifetime of a process.[19]

1.1.3 Role of Good Manufacturing Practice

Good manufacturing practice is the most important collection of guidelines for the production of a drug and the recording of the document to guarantee the manufacture of safe products. The collection is the translation in practical term of the regulations of various countries (21CFR for the United States)[20] and covers practices along with the entire process from the behavior of the employees, to the maintaining and controlling of plant hygiene and product quality. The adherence to the GMP guidelines ensures that the features of each product, e.g., lot number, age, and process parameters, are traceable; hence, it guarantees that the authorities can control the quality of the product on the market. GMP guarantees that the products are not altered during the timeframe from the raw material supplier to the patient.

For improving the practical understanding, some simple examples of GMP guidelines based on the United States Code of Regulation are presented below.[20]

- §211.25: "*Each person engaged in the manufacturing […] shall have education, training, or any experience […]*"

Each employee requires training before starting the operations; therefore, an adequate training program with certification has to be provided to the employees.[20]

- §211.113: "*Appropriate written procedures, designed to prevent microbiological contamination of drug products purporting to be sterile, shall be established and followed. […]*"

Written procedures are defined to maintain standardized operations, the steps of which are always traceable, to maintain the safety of a drug. The practical application of this guideline incurs high cost and is time consuming for the industry; however, it is essential for the patient health to guarantee that all drug produced have the same safety conditions.[21,22]

- §211.188: "*Batch production and control records shall be prepared for each batch of drug product […].*"

Additionally, the standard operating procedures and indicators, which prove the quality and safety of the product, are created, validated, and filed with the authorities.[23]

- §211.110: "*Valid in-process specifications […] shall be consistent with drug product final specifications and shall be derived from previous acceptable process average and process variability estimates […].*"

Such indicators can vary in terms of physical properties, e.g., reflexing index, to process related parameters, such as maximum pressure at a specific process step.[24]

- §211.165: "*Acceptance criteria for the sampling and testing conducted by the quality control unit shall be adequate to assure that batches […] meet each appropriate specification and […] quality […].*"
- §211.192: "*Any unexplained discrepancy […] or the failure of a batch […] shall be thoroughly investigated […]. A written record of the investigation shall be made […].*"

Unexpected events, i.e., deviations or discrepancy, such as the systematic trend in indicators or product not matching the quality, are first documented and subsequently subjected to root cause analysis to identify the reason of such a deviation.[25]

As it has been exemplified, pharmaceutical processes are highly controlled by GMP for the previously mentioned reasons. This high control decreases the degree of freedom towards change with two added risks: First the product quality can be influenced by process changes that seek performance improvements (see

§211.110 above); second, once a process is changed, the entire documentation influenced by this change has to be adapted, which is both time and cost intensive. The lack of degree of freedom, which could be found in chemical manufacturing, renders pharmaceutical companies reluctant to change without a GMP compatible proof—i.e., experimental and statistical testing—of the change. Besides the obvious benefits of such a system, including patient priority, control, and traceability, the sub-optimal process design, as mentioned in section 1.1.2, and the process stiffness do not facilitate process improvement. Sugiyama and Schmidt (2013) developed a model for continuous process improvement, which targets the management of information during the improvement considering GMP.[26,27] Several authors reflected on including the concept of GMP in their studies;[27,28] however only a few incorporated the presence of GMP as a constraint to evaluate its quantitative influence on the process.[29] Such perspective is critical in developing a tool for supporting pharmaceutical manufacturing because GMP and its culture are the limiting factors in changing manufacturing processes.

1.1.4 Opportunities and challenges of pharmaceutical manufacturing

To summarize the current situation outside academia offers various opportunities for improving the performance of manufacturing plants; however there are many more challenges. The fact that the society is aging and the R&D costs are increasing calls for an improvement of the profitability/performance of the manufacturing operations. The non-decreasing cost of production and the sub-optimal design of processes encourage the research of novel methodologies for designing better processes from the root and for improving the performance of the process that is currently running. The incorporation of industrially relevant limitations, such as GMP, in this work, is the key to creating methodologies that are applicable in the manufacturing environment. Finally, it is imperative for guaranteeing the applicability of the methodology that it is understandable and compatible with the culture and the standard operating procedure of pharmaceutical manufacturing.

1.2 Process Systems Engineering (PSE)

1.2.1 Recent development of PSE

Part of the PSE community has always been interested in the decision support for improving processes in various industrial sectors. Bonfill et al. (2008) presented a framework for supporting the coordination of the supply chain management in scheduling problems connecting production and transport in chemical manufacturing.[30] Later, Lainez et al. (2010) holistically modeled the linkage between marketing and supply chain for improving business strategic decisions, and demonstrated the economic benefits of such an approach in a conceptual case study.[31] Suresh et al. (2010) presented a conceptual framework for the ontological and mathematical modeling applied in the knowledge management of pharmaceutical product development.[32] In recent years, another part of the PSE community has shifted its research focus from the modeling and simulation, optimization and scheduling of chemical processes,[33–35] which is still prevalent, to varied applications, such as biomass and energy. Pirola et al. (2017) presented a simulation of a biomass-to-liquid plant and proposed a cost-optimal reactor staging;[36] Martin and Grossmann (2017) proposed an integrated facility that exploits CO_2 capture, for producing methanol from switchgrass.[37] As for the production of energy, Li (2017) analyzed the effect of multi-stage design in the power generation in pressure-retarded osmosis;[38] Elsholkami et al. (2016) developed a model for the integration of renewable technologies in the energy infrastructure in sands industry.[39]

From the works published in the last few years, it is notable that the community is moving from the applied mathematics modelling[40] and optimization[41] to more application-driven research. An example is Boukouvala et al. (2011), who discussed the need for data-driven modeling in a field such as powder manufacturing, to substitute first principle models, which are computationally very expensive.[42] Several studies have been published on the introduction of computer aided-tools for decision-making and support[43,44]; the trend is also to move, through more frequent industrial collaborations, toward more data-driven research in an attempt to understand and solve problems with real-world data.[45–47]

1.2.2 Application of PSE in pharmaceutical manufacturing

Figure 1.3 shows the number of paper over the last 20 years mentioning the keywords "process," "simulation," "modeling," or "optimization," as well as "*pharma*," "API," "drug," or "medicine"; all the papers were categorized as "chemical engineering" from *web of science.*[48]

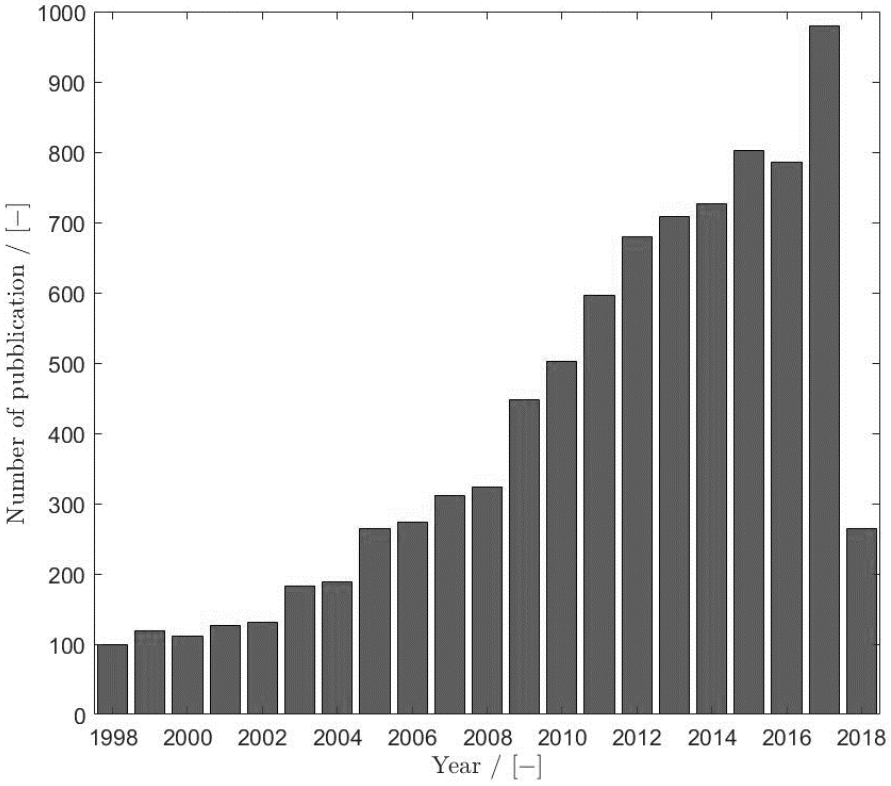


Figure 1.3 Graphical review on the number of papers published in PSE for pharmaceutical application from 1998 till date (14.04.2018).

The number of publications increased over the last 20 years. Especially after the prospective analysis from Reklaitis in 2007 on PSE for pharmaceutical process development,[49] the research exploded to topics, such as molecular design[50,51], sustainability assessments for bioprocesses,[52] development of continuous processing technologies,[53–55] and process design and improvement of operations in both the DS and DP fields.

Various PSE approaches, such as computational process design and process retrofitting, have been applied to achieve the optimal design in the synthesis of pharmaceutical processes as a part of the Quality by Design (QbD) concept. In a review, Rantanen and Khinast (2015) asserted the importance of QbD-based approaches in the pharmaceutical environment and listed various methods that are essential for process modeling.[56] Sajjia et al. (2017) investigated the compaction of microcrystalline cellulose through mathematical modeling, for

designing of an industrial-scale roller compaction process.[57] Casola et al. (2015) presented a Modeling–Simulation–Optimization (MSO) methodology for retrofitting the recrystallization process of injectable drugs.[58] The first principle MSO approaches tackle the process design, synthesis, and optimization, whereas data science approaches are used in process assessment[59] and control[60], as well as in the improvement of operations[61].

Regarding DS, Papadakis et al. (2017) created a reaction database, in order to collect multiphase reaction data to be used in the reaction-separation system, for small molecules processes.[62] Diab and Gerogiorgis et al. (2018) proposed a techno-economic evaluation of the purification of rufinamide by antisolvent crystallization from the modeling and simulation of the upstream synthesis process.[63] A mathematical modeling and optimization approach for a continuous multi-stage slug flow crystallization process was shown;[64] Ridder et al. (2016) presented a feasibility study of utilizing a multisegment antisolvent crystallizer for fines removal and matching of the target crystal size distribution.[65] Luo et al. (2018) reviewed the recent advances and strategies in process engineering of the microbial fermentation of butyric acid, which is an important platform chemical for the pharmaceutical production[66]. An optimization that considers uncertainty, of upstream and downstream processing for biopharmaceutical manufacturing, was proposed by Liu et al. (2016).[67]

As for DPs, Martinetz et al. (2017) developed an operating concept for a rotary tablet press that uses mass flow operating point.[68] Sajjia et al. (2017) proposed an artificial neural network for simulating the process of dry granulation via roller compaction.[69] Içten et al. (2016) showed the application of the Knowledge Provenance Management System for modeling the relationship of processing steps, materials, and information as an innovative analysis of experimental data from drop-wise additive manufacturing.[70] Furthermore, for continuous solid-dosage manufacturing, Su et al. (2017) presented a systematic framework for the process control and business risk analysis.[71] Among the few studies that deal with biopharmaceutical DP manufacturing, Bosca et al. (2015) proposed a risk-based design modeling for freeze-drying cycles compliant with GMP regulations.[72] Eberle et al. (2016) presented a data-driven tiered procedure for enhancing production yield; the procedure was applied to a sterile filling process in an industrial case study.[73]

A more specific literature review on particular application of PSE approaches in the pharmaceutical field is found in the introduction of **Chapter 3**, **4**, and **5**. The additional literature review is used to highlight the academic surrounding specific to each chapter.

1.2.3 Opportunities and challenges for PSE in pharmaceutical manufacturing

The incorporation of PSE approaches in designing, controlling and changing pharmaceutical processes provides excellent opportunities for improving the current industry. Topics, such as performance improvement, reliability assessment, and process control and improvement can be well supported by the PSE and computer-added approach as prominent authors in the field mentioned in their reviews.[49,74] Challenges are found in the implementation of process changes in running plants because of GMP, which requires experimental results proving statistically significant statements. Being both an opportunity and a challenge, the availability of real-world data will be vital in determining the impact of the research on the field of manufacturing, dividing hypothetical/ideal from verisimilar models. Drug product manufacturing has proven to greatly impact the all drug production, therefore it is imperative to consider this step when assessing and improving the performance of the entire manufacturing process. In addition, from the literature review, it was recognized that the fields of small molecule DPs and DS, as well as biopharmaceutical DS, are well covered; however, significantly fewer studies on biopharmaceutical DPs were found, which opens the opportunity for new research.

1.3 Digitalization and advanced applied statistics

1.3.1 Digitalization in various industries

Among numerous definition of digitalization, one by Gartner states that it is "*the use of digital technologies to change the business model and provide new revenue and value-producing opportunities*".[75] The term "digitalization" differs from the already existing "digitization"; in fact, the first refers to the use of the technological development to disruptively change the current business models in a data-driven revolution, whereas the second refers to the technical switch from analog to digital data recording and storage. Process and business data are not the only elements to be digitized; digitization of workflows, processes, and documentation is a requirement that sets the basis for pursuing the digital revolution.[76,77] It is of common knowledge that various industry sectors started changing their business models through digitalization; internet companies based on supply chain systems, such as Amazon and Google, reached the leadership of the market by implementing digital technologies. Further concepts, which go hand-in-hand with digitalization, are Industry 4.0 (I4.0) and

Industry of Things (IoT); this section displays some academic works that show efforts for the digital revolution in various industrial sectors. In addition to the implementation of advanced analytics and Internet of Things in the creation of a wide digitalized and patient-centric healthcare system, such technologies are promising in the improvement of reliability and efficiency of manufacturing processes. In a review, Volker et al. (2016) presented the current trends towards digitalization and the use of big data analytics in the healthcare system mentioning several advantages, including at-risk patient identification or hospital readmission prediction.[78] Industry sectors, such as banks,[79,80] and automotive[81] have been embracing the digital revolution for years and changing their business models. Tao et al. (2011) reviewed the application of grid technology in manufacturing systems.[82] More recently, Blaya et al. (2018) designed an arm splint, which is an orthopedic product, by using additive manufacturing technology.[83] Reitze et al. (2018) presented a roadmap for the so-called Smart Factory, which includes the modular manufacturing concept for the flexible production of specialty chemicals.[84] In a recent review, Chiang et al. (2017) showed the recent advancements of big data technology in various industries from a chemical engineering stem point; applications of big data in the pharmaceutical sector were also presented, where most of the efforts are seen in the field of drug discovery and design. The progress in the fourth industrial revolution in the pharmaceutical manufacturing sector is still at an infant stage; however, some studies are found in the literature. Lepage et al. (2013) proposed a real-time method that uses semiconductor photonics, for detecting the influenza A virus;[85] Schenkendorf (2016) stated that the shift from batch to continuous in pharmaceutical manufacturing can be supported by data-driven prediction models, which can control, optimize the processes and predict faults.[86] To highlight the specific status of the research and the needs of the pharmaceutical industry, a more detailed academic literature review on the topic is found in **Chapter 5**.

1.3.2 Opportunities and challenges for "data science" in pharmaceutical manufacturing

Currently, most of the effort toward digitalization is contributed by the industry; however, studies that are more introspective are needed to analyze and discover the real potential of digital technologies in improving processes. A major challenge in pursuing this path comes with redefining the concept of innovation since most of the "data science-" approaches are 20 years old. Application-driven studies and industrial collaborations are required to provide implementable methodologies that can support the manufacturing of pharmaceutical

manufacturing. The presence of GMP plays a key role in both reducing and enhancing the speed of the change; in fact, the stiffness of GMP not only protects state-of-the-art processes but also obliges to record data, which are an essential asset for this innovation. Employing "data science" approaches in the current pharmaceutical manufacturing could lead to redefining some of the very traditional principles and as QbD or process control through validation. A practical example of the effect that the digital revolution is the capability of predicting process faults in very complex systems, which would result in reducing downtime and costs.

1.4 Process targeted in the case study

1.4.1 Definition of the process

The cleaning-in-place/sterilizing-in-place process, which is utilized in all industrial case studies in the thesis, is employed as the change-over operation of a sterile filling plant belonging to F. Hoffmann-La Roche in Switzerland. The plant showed in **Figure 1.4** is responsible for filling an API solution, the API usually being an mAb, in glass vials.

The filling plant consists of a vial washing machine, a sterilization tunnel, a filling and a capping system, and an inspection machine (not visible). The filling and capping systems are located inside the isolator as shown in the schematic representation in **Figure 1.5**. As it is shown in **Figure 1.5**, first, the vials are loaded into a washing machine, where dust particles are removed by spraying the internal and external surface of the vials with Demineralized Water (DW).

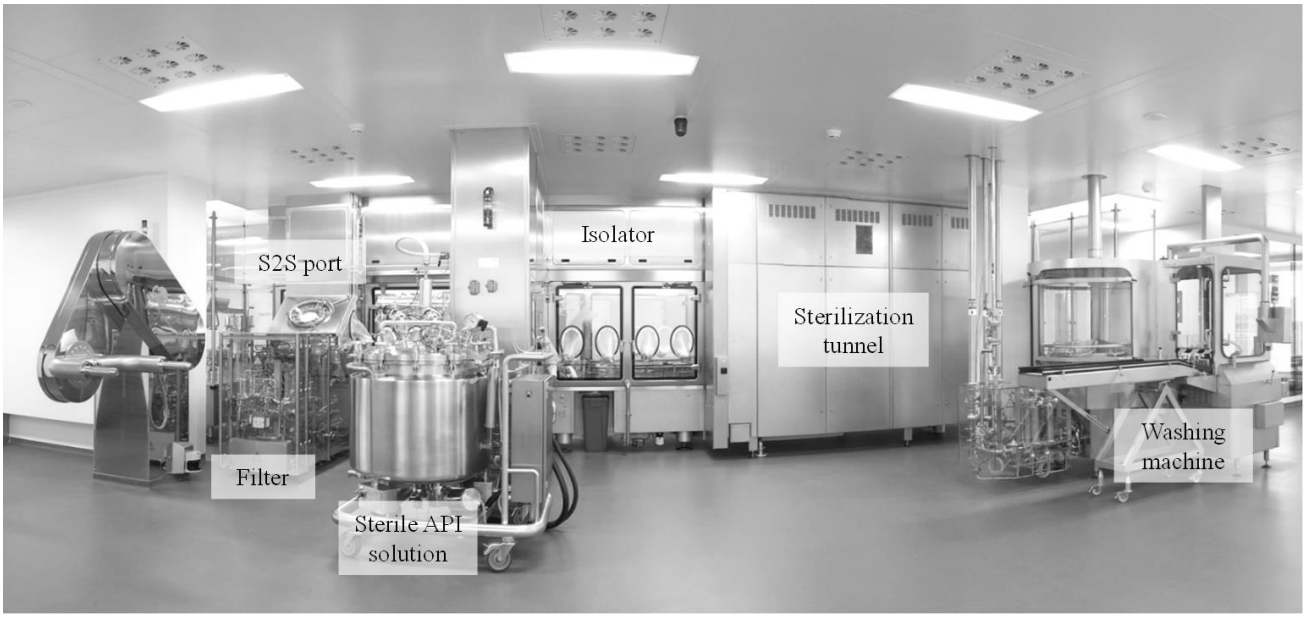


Figure 1.4 Picture of the plant that belongs to F. Hoffmann-La Roche Ltd. in Kaiseraugst, Switzerland

Second, the vials are automatically transported into the sterilization tunnel (see **Figure 1.6**, A), in which a temperature of 300°C activates the decomposition of pyrogenic components such as introns, bacteria, or viruses; after sterilizing for 10-15 min, the vials enter the isolator. Third, the sterile vials are filled with the sterile API solution, which passed through a filter (see **Figure 1.4**) before being transferred to the buffer tank in the isolator; a multi-needle filling system, which is connected to the buffer tank via plastic filling tubes, fills 6–12 vials simultaneously (see **Figure 1.6, B**). Fourth, the vials, whose weights are checked sample-wise, are sealed with pre-sterilized stoppers (see **Figure 1.6**, C), capped (see **Figure 1.6**, D) and are transported outside of the isolator. Fifth, the vials are visually inspected by a system of 13 cameras; the vials that show defects such as scratches, air bubble inclusions in the glass and particles inside the product are discarded.

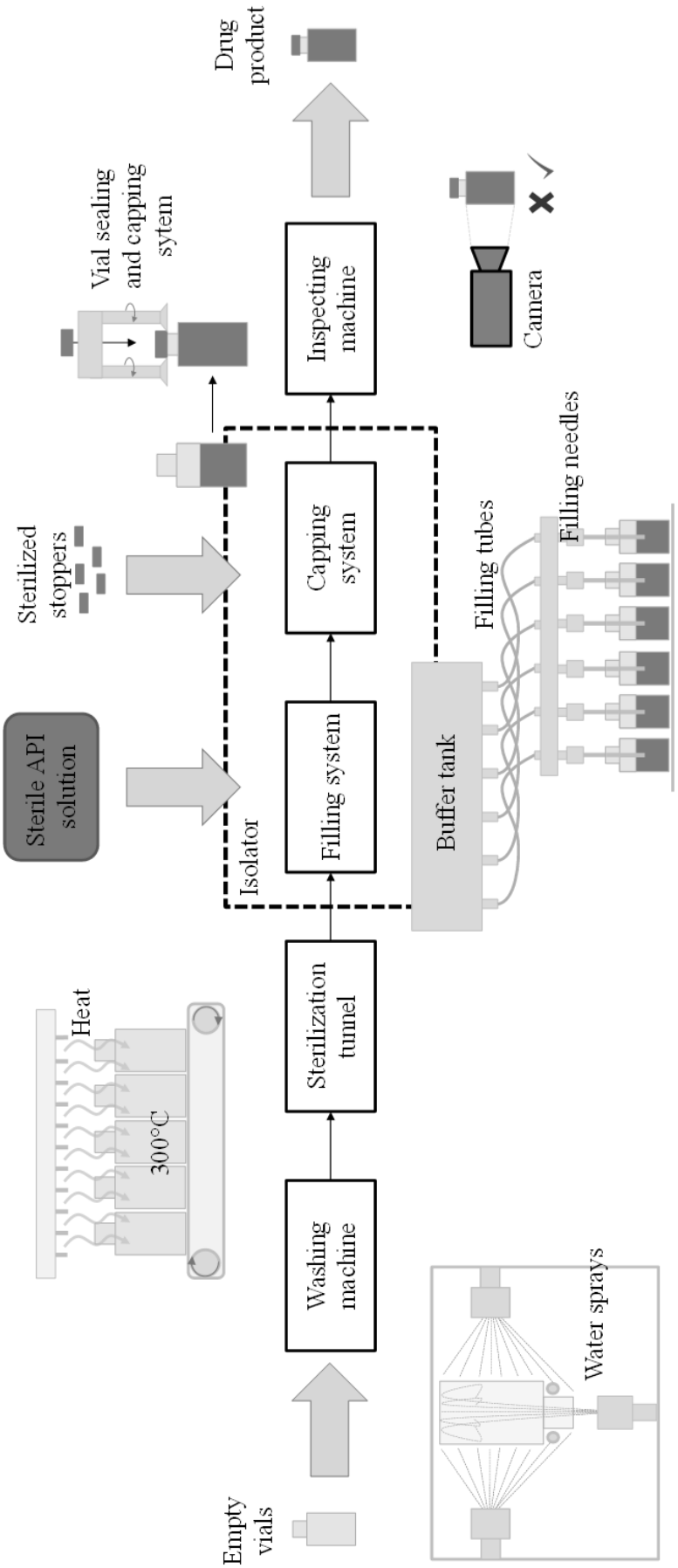


Figure 1.5 Graphical representation of the drug product manufacturing process.

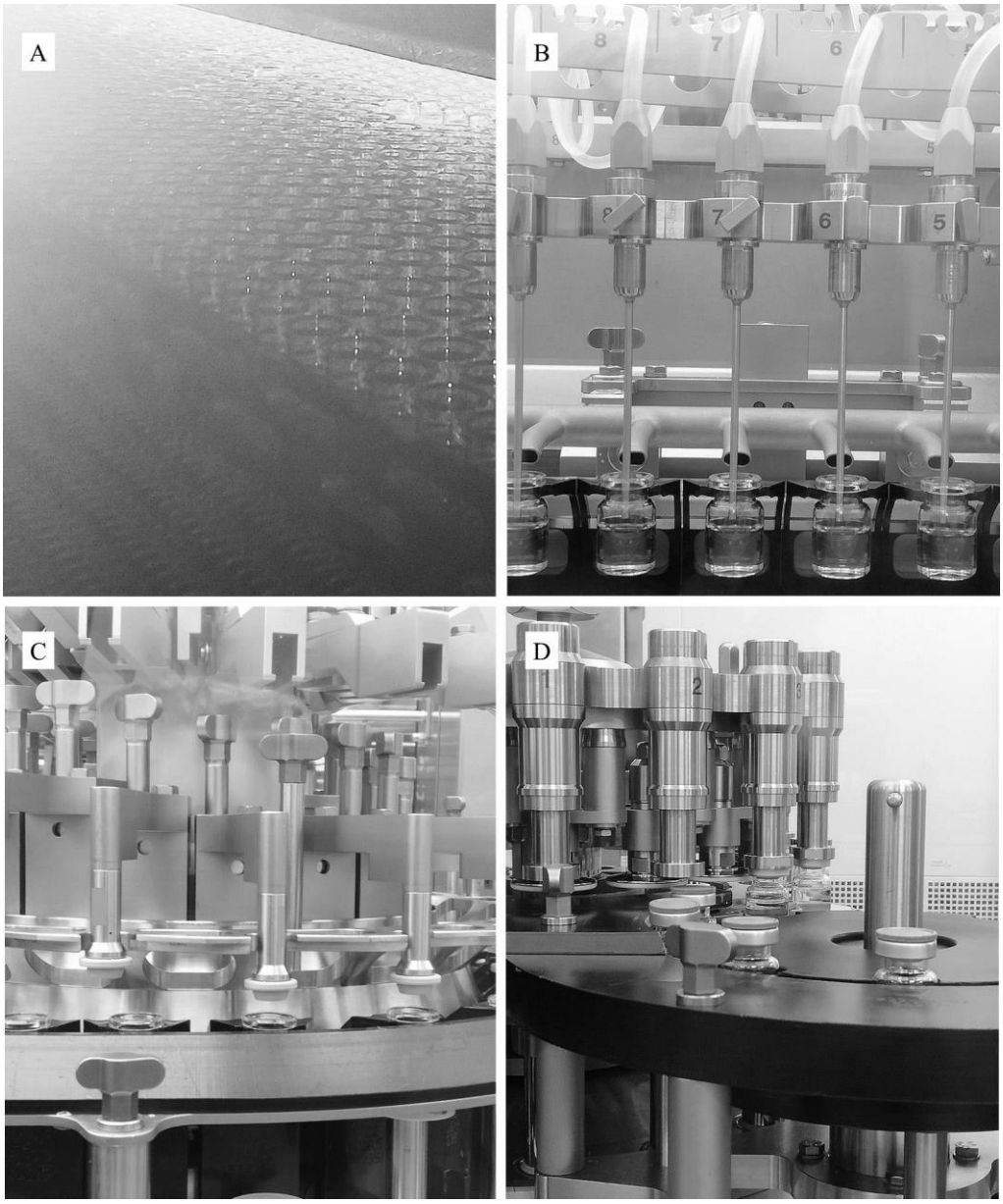


Figure 1.6 Pictures of the filling equipment: sterilization tunnel (A), filling needles (B), stoppers sealing system (C), and capping system (D)

As previously mentioned, before each filling batch, change-over operations, namely the CIP/SIP process, is required. The investigated commercial CIP/SIP process is applied to the filling system depicted in **Figure 1.7**. The filling system, shown in **Figure 1.7**, consists of a filter, a buffer tank, single-use plastic filling tubes and filling needles; the filling system is attached to a utility supply system that supplies DW, Water for Injection (WFI), and Pure Steam (PS) and to the tank containing the API solution for one filling batch. At each filling batch, the filling tubes are manually exchanged, and the API solution tank is attached to the filling system by a

sterile-to-sterile (S2S) port (also see **Figure 1.4**), which guarantees the sterility of the filling system and the solution.

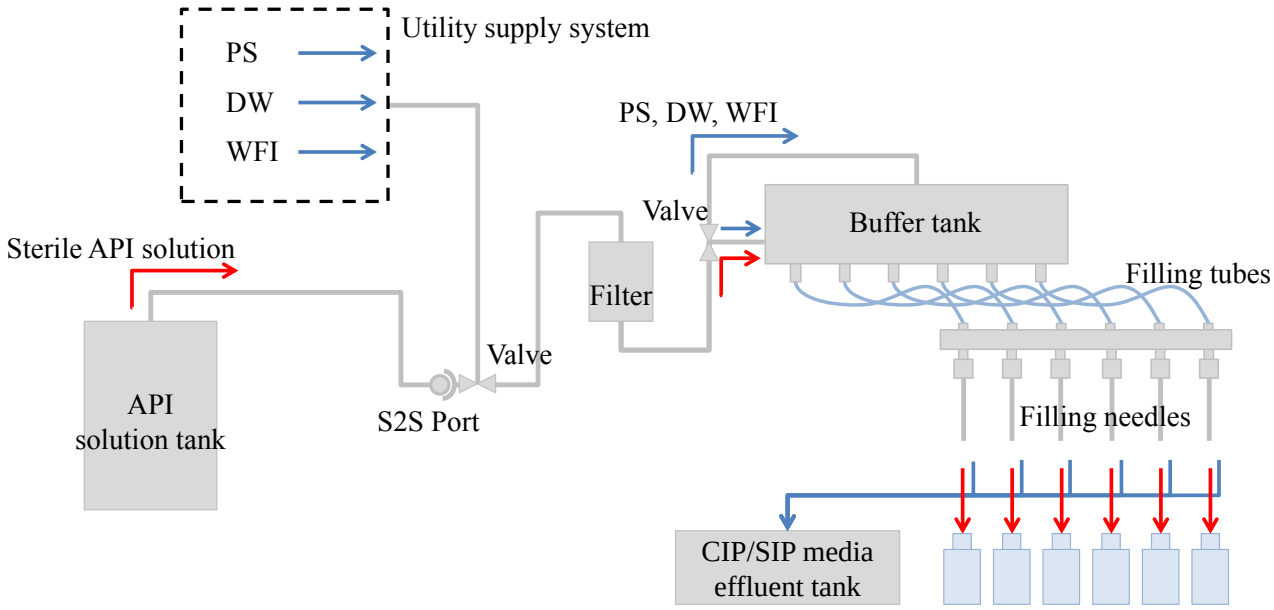


Figure 1.7 Simplified graphical representation of the sterile filling operations; CIP/SIP (blue arrows) and API filling (red arrows).

The valves shown in **Figure 1.7** are set towards the blue arrows whenever utilities, also called CIP/SIP media, are required, namely during the CIP/SIP process; during the filling operations, the valves are directed toward the red arrows letting the API solution through the filtration system and the filling needles into the vials. The CIP/SIP medium flows through the entire pre-built piping system and is discharged in the effluent tank, whereas the solution tank and S2S are pre-sterilized and attached to the piping after the CIP/SIP process is concluded.

The CIP/SIP processes consist of 180–252 consecutive tasks divided into 9-12 process blocks. In this case study, the production mode—i.e., the production campaign with multiple batches of multiple products or formats—requires the execution of three different CIP/SIP processes: pre-campaign (252 tasks), intra-campaign (186 tasks), and post-campaign (180 tasks) CIP/SIP. **Figure 1.8** shows the three CIP/SIP process recipes, which comprise the blocks rinsing the piping with DW, testing the integrity of the filters, testing the impermeability of the system, rinsing the piping with WFI, sterilizing with PS, drying and testing the integrity of the product filter system (light gray boxes in **Figure 1.8**). The three CIP/SIP processes are similar but with difference existing in the presence or absence of particular blocks (dark gray boxes in **Figure 1.8**). Specifically setup

operations, which involve the manual mounting of format-specific equipment components, decontamination of the isolator, and integrity testing of the gas filter—i.e., filter used to ensure the purity of the steam—are present only in the *pre*-CIP/SIP; similarly, format change, which involves the adaptation—i.e., mounting and unmounting of equipment componets—of the filling plant to a different format of vials, and the exchange of the filling tubes, in sterile conditions, is present only in the *intra*-campaign CIP/SIP.

The CIP/SIP tasks are operated under the controls of execution times, pressure, and temperature; in specific, all tasks have upper and lower control limits of pressure and temperature as design specifications. Some tasks, such as impermeability and integrity test, are controlled by target temperature or pressure, and others, such as rinsing and decontamination, are limited by time. The control parameters, namely, minimum, maximum and target, are set during the validation and qualification of the process and the equipment, respectively; they ensure the reproducibility of the process and guarantee the achievement of the product specification at each batch.

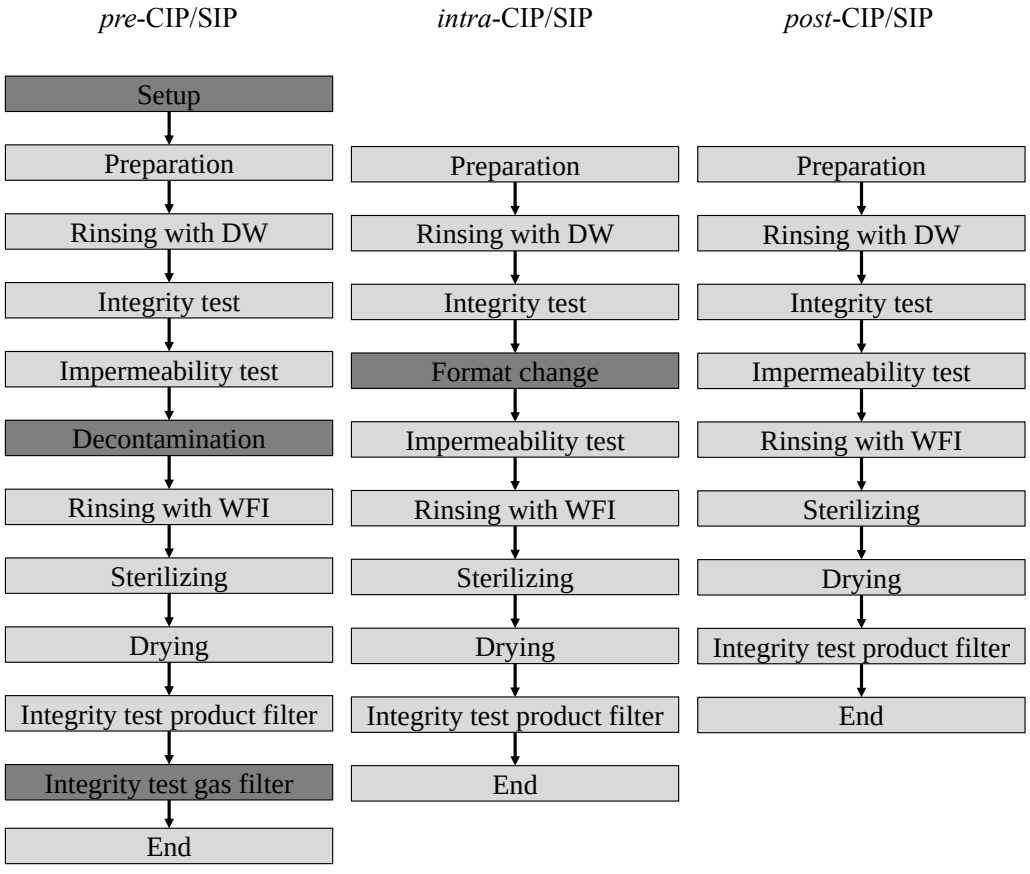


Figure 1.8 CIP/SIP process recipes

All CIP/SIP processes are influenced by human intervention because of the multi-product/format production mode; in fact, mechanical elements used for transporting the vials, e.g., plastic transport screws and vials pincers, are format specific and part of the filling equipment, i.e., the single-use plastic tubes connecting the buffer tank and the filling needles, is product-dedicated and single-use and needs to be manually substituted before each filling campaign/batch. The process is influenced by uncertainty; in fact, each CIP/SIP task has a probability to fail.

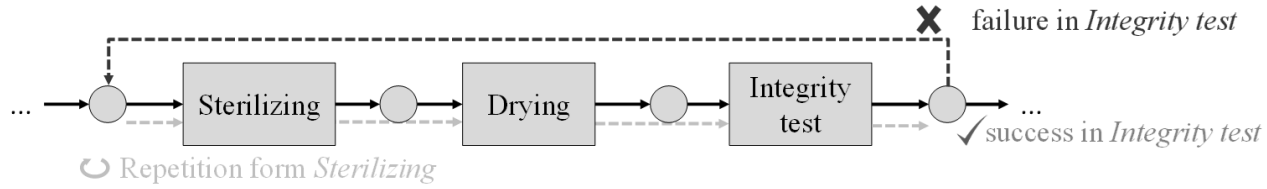


Figure 1.9 Process execution showed task *Sterilizing*, *Drying* and *Integrity test*. *Sterilizing* must be repeated in the case of failure.

As shown in **Figure 1.9**, in case of a failure a specific series of tasks is repeated, with the repetition sequence defined by the GMP documentation; the repetition is necessary to guarantee that conditions such as sterility and tightness are ensured after the failure.

The process produces principally three types of data, namely, sensor data, execution logs, and metadata; irrespective of whether a specific process is running, the data are continuously recorded and archived in data storage units. As for the sensor data, the plant possesses five pressure [bar] and nine temperature [°C] sensors, the outputs of which are recorded at a frequency of 1 Hz. The execution logs are the temporal sequence of the task performed during the operations; at the start of each task, a time stamp (raw in **Table 1.1**) consisting of time and task ID/task description is recorded in text form. A graphical representation of the execution log is presented in **Table 1.1**. Metadata is a mixture of text and numerical data, and usually, it contains information regarding time logged process commentaries, such as logbooks, process comments, and background information, such as operator ID.

Table 1.1 Example of an execution log

Time	Task ID	Description
04:07:13 15.03.2018	'3000'	*Set media pressure in the piping*
04:07:33 15.03.2018	'3010'	*Test media pressure in the piping*
04:09:33 15.03.2018	'3020'	*Increase pressure in the filter unit*
04:11:33 15.03.2018	'3030'	*Test leak-tightness for 2 min (pass if* $\Delta p < 0.2$ *bar)*

1.4.2 Problem framing

The following characteristics were noticed by analyzing the current processes and operations standards in the decision-making in the example of the CIP/SIP process:

The decision-making in the pharmaceutical industry is based on scientific discovery, experiments, i.e., from process design to scale-up, on rationales, which deterministically connect events through logic or speculation, and on experience. However, it does not consider uncertainty. An endogenous uncertainty event, e.g., operator mounting mistake or machine failure, is an omnipresent part of the manufacturing operations but not part of the decision-making. In batch processes such as the CIP/SIP, where the operators interact with machines, e.g., while mounting and unmounting plant parts, failures can occur; hence, uncertainty has to be considered. Examples of failures observed in the studied process comprise leakage trough gaskets, wearing or wrong assembly of equipment components, and mechanical failures, such as malfunctioning of robot arms.

Other factors that are peculiar to the case study compared with those presented in the literature are the facts that the process is highly complex—i.e., first principle modeling is not feasible—and it does not give the opportunity of experimentation. The collection of new data by installing additional sensors (if possible) would require years to reach a dataset that carries a statistical relevance. Hence, the historical data provided by real plant cannot be expanded. Similar to many commercial processes, the production facility is not available for performing experiments because the timeline is saturated with commercial batches. Given these two factors, decisions can only be performed on the basis of the historical data recorded during commercial batches and their interconnection. The data-driven character of the decision-making is specific for pharmaceutical processes, but

can be advantageous in all industries—i.e., less experimentation translates to more production; therefore, it has to be incorporated in the study.

The last characteristic is GMP; in industries, implementing GMP is crucial whenever process are involved; therefore, it has to be incorporated in the decision of making changes in the process. Extensive analysis of the challenges and opportunities when working in a GMP-controlled environment has been presented in section 1.1.4.

Chapter 2: Objective of the study

2.1 Framework for the decision-making in process improvement and operations support

In this study, a framework is presented, whose application would assist the uncertainty-conscious and data-driven decision-making in the process improvement and operations support in biopharmaceutical manufacturing. The framework, shown in **Figure 2.1**, is centered on the process (green box); the manufacturing process delivers raw data. The raw data are preprocessed by an automated algorithm (white box, **Figure 2.1**), which uses past human decisions taken during a conventional manual data cleaning, and language recognition algorithms to train classification models. The preprocessing algorithm first removes the data that do not carry process information, second, it clusters the data into single batches, and finally, it labels the data points by their process characteristics, i.e., alarm, repetition, and process data.

The clean data are employed either to advise process re-design and debottlenecking alternatives (left blue box, **Figure 2.1**) through the assessment of the process performance, such as runtime (left gray box, **Figure 2.1**), or to plan preventive or corrective maintenance actions (right blue box, **Figure 2.1**) through the prediction of imminent failures (right gray box, **Figure 2.1**) during the manufacturing operations. The first route uses historical runtimes as sampling pools for the stochastic sensitivity analysis considering the uncertainty of operations with the objective of identifying process tasks that limit the total process performance. The second leverages principal component analysis and supervised machine learning to extrapolate hidden process features from historical sensor data aiming at characterizing process failure. The goal of the second route is to identify process trends that would result in failures, namely, unexpected failures, before their occurrence and in real time; the prediction model transforms unexpected failures into predictable or even preventable failures, thereby reducing unexpected downtime. The outcomes of both routes are implemented in the manufacturing process; the framework is tailored recursively to enable the continuous improvement of the manufacturing without performing experiments until reaching process optimality—i.e., minimum runtime and downtime.

In addition, to provide a guideline for continuous process improvement, the framework can be employed as a computer-aided tool for the real-time and uncertainty-conscious decision-making which is employable in commercial-scale manufacturing operations. Each activity of the framework intakes the input of experts, such as engineers, operators, and analysts, which are required to translate their knowledge into quantitative and semi-

quantitative indicators and of data records, which set the basis for the decision. The tool can be tailored to various industry sectors; however, in this study, it will be employed in the drug product manufacturing sector.

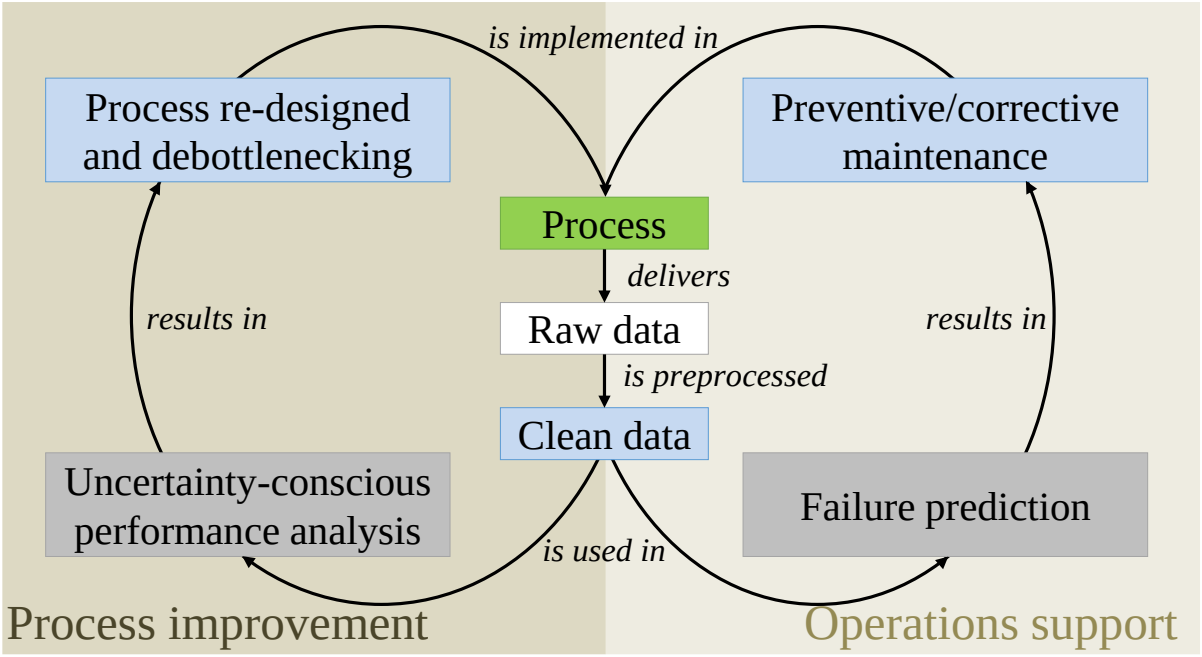


Figure 2.1 Framework for the decision-making in process improvement and manufacturing support

2.2 Thesis statement

The thesis presents a computer-aided framework for the assistance of data-driven decision-making for process improvement and operations support in biopharmaceutical drug product manufacturing. Commercial assembly-like processes are the target of the framework; the framework intakes historical data and expert knowledge to deliver an improvement solution suited to the manufacturing environment and industry sector. **Figure 2.2** shows the thematic location of the framework, which integrates knowledge from the various research fields, namely, pharmaceutical, PSE, and digitalization, to deliver tailored decision-making. The intersection between these research fields, especially with digitalization position the research in a very new and innovative research field, which is committed to developing highly applicable tools integrating the new state-of-the-art technologies.

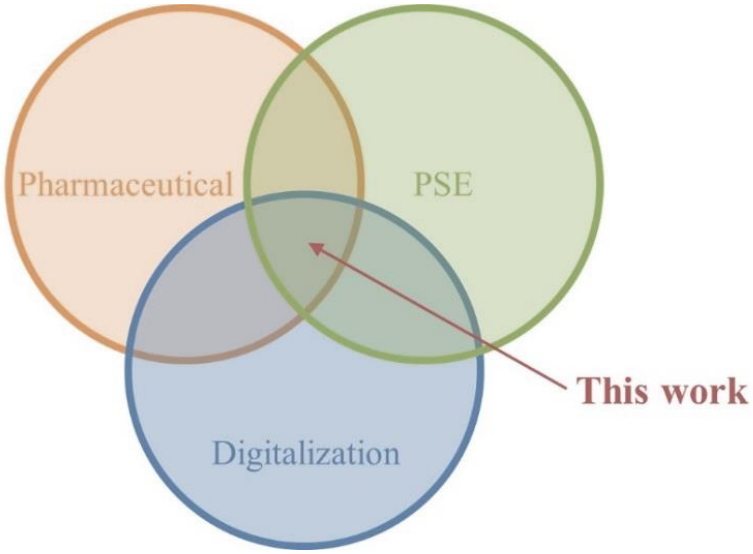


Figure 2.2 Positioning of the framework in the current research landscape

The following points are the general concepts necessary for the construction and implementation of the framework:

- Compatibility to industrial IT and manufacturing system through general and automated data transformation algorithm.
- Utilization of historical data for fitting specific descriptive or predictive models for each process or task.
- Incorporation of process and operations-specific characteristics, such as pharma-specific limitations and uncertainty to provide tailored decisions.
- Translation of industrial experts' knowledge in to quantifiable indicators and supply of an interface that is understandable for non-experts for the implementation in the industrial environment.
- Supply of a tool that supports and guides the human decision by considering risks of a process modification or intervention.

The following are the goals that are necessary to satisfy the personal interests of the author:

- Incorporation of industrial, information, digital, and process system engineering techniques to provide multidisciplinary and multi-faceted research.
- Bridging the gap between academic and industrial tools for providing higher value results through innovation by incorporating advanced analysis methods into conventional frameworks.

The following are the achievements in implementing the framework in a commercial environment:

- Automation of obsolete tasks, such as data cleaning, based on historical human decisions.
- Delivery of an uncertainty-conscious decision that considers the human influence on the process performance.
- Introduction of advanced statistical methods for the extrapolation of event-specific features.
- Reduction of unplanned downtime by predicting imminent failure
- Introduction of Industry 4.0 and big data concepts in the biopharmaceutical manufacturing.

2.3 Thesis structure

The thesis is structured as shown in **Figure 2.2**. **Chapter 3**, **Chapter 4**, and **Chapter 5** present the data preprocessing algorithm, the performance assessment methodology, and the failure prediction algorithm, respectively. Each chapter has an independent introduction, conclusion, and nomenclature sections. The outcome of **Chapter 3**, namely the clean data, is connected to both **Chapters 4** and **5** in a parallel structure; the order of **Chapters 4** and **5** is chronological of the conclusion of the works.

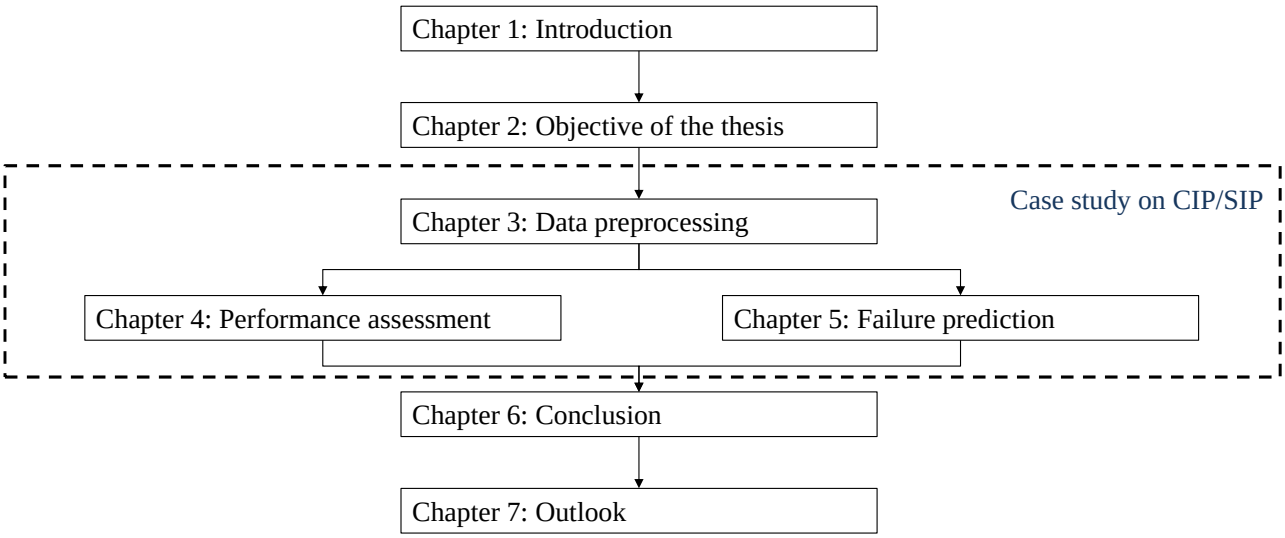


Figure 2.3 Structure of the thesis.

Due to the direct incompatibility of the data records with work presented in later chapters, **Chapter 3** introduces an automated algorithm for the transformation of raw data into clean data that can be used for further analysis. First, the algorithm is defined; after the homogenization (stemming) of the dataset, the raw data, which are a continuous strain of points in the form of strings, are sequenced like a DNA sequence. The resultant sequence is then used in the identification of noise with supervised machine learning models; subsequently, the

noise is filtered and the single data points are characterized. Second, the algorithm is designed for the case, namely the parameters that deliver the maximal accuracy in noise classification are selected, and is tested for stability over time. Third, the algorithm is applied to the case study where the conventional manual preprocessing was used as the input knowledge to train the noise classifier and as a reference. Fourth, an alternative utilization of the data is proposed, namely, the connection of the data with the root-cause analysis performed in case of process failure; consequently, two innovative quantitative root-cause analysis representations are delivered. Finally, the chapter is concluded with a discussion on the improvement resulting from the implementation of the algorithm in the manufacturing scale.

Chapter 4 presents the methodology, in the form of an activity model, for the uncertainty conscious performance assessment leveraging stochastic global sensitivity analysis. First, a six-activity model (IDEF0 model) is defined; one activity (activity A3) of the model is connected to **Chapter 3**; the model takes manufacturing data and identifies the process task that has the most influence on the total process performance. Second, the activity model is applied in an industrial case study objecting the CIP/SIP process. In this case study, the methodology provides an answer to the question: "Knowing the operators are an omnipresent source of uncertainty, which task is the bottlenecking task?" The chapter continues by quantitatively showing the outcome of the process performance of potential process changes through what-if analysis. Last, the chapter closes with a conclusion where the results of the presented methodology are compared with those resulting from the application of an industrially conventional method.

Chapter 5 introduces the development and application of an intelligent algorithm, which is based on Industry 4.0 and big data approaches, to the pharmaceutical manufacturing industry. First, the algorithm is defined, which predicts the failure status of the process from physical sensors in real-time. A retraining loop maintains the quality of the prediction over time because the algorithm is based on machine learning and the process is in continuous evolution; the algorithm results in a decision after the analysis of the risk on the performance in case of action. Second, a case study is presented where the algorithm is used to predict failure and then to support the human decision of taking either a preventive or corrective action. Events describable through first principle models or empirical models are simulated to analyze the capability of the algorithm to recognize unknown imminent failures. Last, the chapter closes with a conclusion where the potential unexpected downtime

reduction is quantified and the implementability of the algorithm in commercial-scale manufacturing is assessed.

The results and limitations presented in the previous chapter are summarized, and a conclusion is drawn in **Chapter 6**. **Chapter 7** presents the thesis outlook; the outlook tackles the expandability of the work to industry sectors different from pharmaceutical manufacturing, the expandability of Industry 4.0 concept inside pharmaceutical manufacturing, and the potential impact of such a work on the environment.

Chapter 3: Data mining algorithm for pre-processing biopharmaceutical manufacturing records

(*Based on the manuscript submitted to Computers and Chemicals Engineering by G. Casola, C. Siegmund, M. Mattern, and H. Sugiyama*)

3.1 Introduction

The DP manufacturing process consists of a series of independent tasks that are executed consecutively by following a process recipe. A process recipe defines the activities, the sequence of activities, and the process parameters used to produce a particular product. DP manufacturing plants produce several products with the same equipment setup, which is finely adapted depending on the process recipe; generally, process recipes have multiple common tasks, but additional ones are present if the products necessitate particular requirement—e.g., a vial with an additional metallic cap or only rubber cap. Because of manual operations such as format changes at the filling line, or nonroutine events such as failures, the process presents uncertainty, which, in this work, is referred to as operational uncertainty. A detailed description of operational uncertainty and its appearances can be found in **Chapter 4**.[29] This uncertainty is the cause of a specific type of disturbance in the data, called "operational disturbance," which appears as a discontinuity in the execution of the process recipe in the dataset (see Appendix: Tutorial for a graphical example).

The recent developments in data science-based technologies opened the field of manufacturing to principles such as cloud- and smart-manufacturing, not only for the refinery industry[87] but also for the pharmaceutical industry[88]. Kemppainen (2107) discussed the challenges and opportunities in transforming the pharmaceutical manufacturing industry by increasing digitalization.[89] Already facilitated by the GMP companies collect a significant amount of manufacturing data presenting great opportunities towards digitalization. Traditionally, GMP regulates and records all the activities in the manufacturing practice of pharmaceutical products[90]; through the tracking of a large quantity of information, GMP aims at ensuring the safety of drugs for the patients. As mention in **Chapter 1**, the essential principles of GMP regarding data are the completeness, consistency, and accuracy of processes and products data, principles that are referred to as *data integrity* in the current GMP (cGMP).[91] Currently, in the industry, practices that use manufacturing data, such as periodic reviews, projects and performance analyses, require extensive investment in understanding and manually tracking raw data—e.g., finding specific process information for a specific batch within the continuously recorded data. It is necessary to have not only reliable databases to guarantee data integrity, but also an automated data reporting system for preventing human errors. The application of data mining techniques has the potential to increase the efficiency of monitoring and to exploit manufacturing data with reduced human interactions.

At state of the art, the pharmaceutical industry faces three major questions in achieving such goals. The first is whether the advanced data-mining knowledge gathered in other industries in the last decade is leveraged efficiently; the second is whether the data currently recorded is qualitatively satisfactory to be used directly for decision-making, and the third is whether the morphology and the data are compatible with the existing batch isolation method, such as the tracking of flagships.

First, the data recorded during production is often only used for monitoring, controlling and sometimes, improving operations. By contrast, in various fields of manufacturing, such as the assembly[92], metallurgical[93,94] and textile[95] industries, data mining approaches have been used extensively. Methods such as supervised and unsupervised machine learning and regression have been used for pattern recognition, classification and other types of knowledge extrapolation. Kusiak (2006) reviewed the advantages and challenges of data mining approaches in various types of manufacturing industries.[96] Ma and Wang (2009) proposed a data mining algorithm for the automatic construction of decision trees that was applied to an analysis of process historical data from wastewater treatment plants.[97] Simon and Hungerbühler (2010) compared the use of various intelligent pattern classifiers for the mining of an industrial batch dryer.[98]

Second, the quality of mining activities is strongly dependent on the data quality. In fact, before applying any intelligent algorithm, the data often requires a preliminary "cleaning".[61] Differently to the operational disturbance, noise, defined in this study as the data points that do not describe commercial processes, is generated by the data collection system, which records data continuously without discriminating whether the data describe a commercial process or other events. Some examples of noise are data points generated during plant testing, maintenance, experimentation and in general noise is data recorded between commercial batches. For this reason, the transformation of raw data—e.g., timestamp, ID, and text in general[91]—into noise-free numerical data is also reviewed in this study. Data in text form, such as a string, are a common type of data used in computational biology, particularly in DNA sequencing[99] and natural language recognition[100]. These studies base their findings on the so-called approximated string-matching algorithm developed by Ukkonen (1985),[101] which uses the Wagner–Fischer (WF) distance [102] of two strings to quantify their degree of similarity. Other applications of the WF distance in pattern recognition can be found, for example, in the work of Bunke and Csirik (1995).[103]

Third, data preprocessing can be a simple task if the data recording systems and dataset morphology provide the opportunity to track the process flag points—i.e., the process start and end; however, if such a function is not composed of the system's functionalities, different approaches are needed to perform data processing in an automated and efficient manner. In this work, the manufacturing processes continuously produce two types of data: physical data and metadata in the form of timestamps consisting of execution time and task ID.[104] Consequent to the previously published study by the author[29], which did not investigate the issue of data collection quality and integrity, the metadata are defined in this study as the raw data.

Despite the abundance of studies in data mining, approaches for transforming raw data into clean data—i.e., data that are suitable for direct analysis—are rarely found in the literature, especially in pharmaceutical manufacturing. Meneghetti et al. (2016) presented in a proof-of-concept study a data mining-based algorithm for recognizing batches and process phases using data historians in DP manufacturing.[105] Various limitations, which are often the case in biopharmaceutical DP manufacturing, were highlighted by the authors in the conclusion of that study. Examples of such limitations are the low classification accuracy in the presence of noisy data signals, the instability of isolating single batches and the unsatisfactory classification results because of a nonadaptive clustering algorithm. Such limitations can jeopardize the implementation of the algorithm in an automated manner, which is desirable. In a previous work of the author,[29] the raw data were preprocessed manually because the morphology of the data did not allow the automation of the procedure, resulting in extensive time investment.

In this study, a new algorithm for transforming raw data recorded in biopharmaceutical manufacturing into clean data in an automated manner was presented. The approach integrates natural language recognition and computational biology techniques, as well as machine learning for the selecting and filtering of noise from the raw data without the filtering of the operational disturbance. The resulting dataset only contains the information required for the decision-making when improving process performance and providing support to the operations considering operational uncertainty. Additionally, the practical use of the clean data in the analysis of process performance through an evolved Lean Six Sigma (LSS) approach is highlighted. The applicability of the algorithm to transform execution logs, the raw data, into the clean data and its usability in a commercial environment are shown in the industrial case study.

3.2 Definition of the algorithm

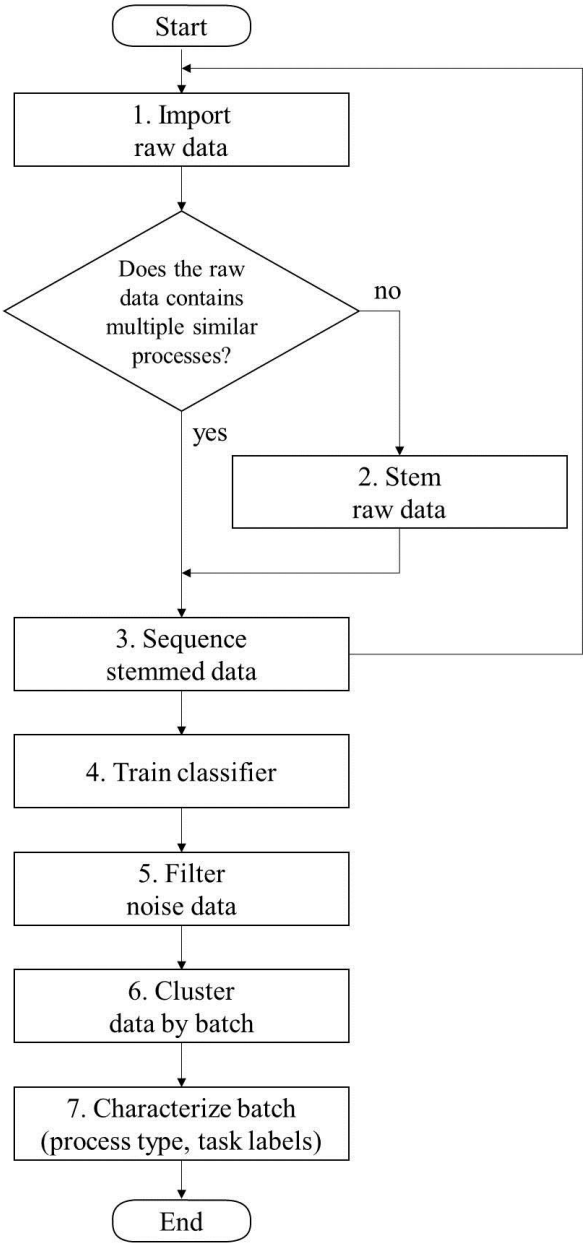


Figure 3.1 Data preprocessing algorithm.

The proposed approach consists of seven steps (see **Figure 3.1**). Namely, import raw data—i.e., execution logs—, stem raw data, sequence stemmed data, train classifier, filter noise data, cluster data by batch, and characterize batch. In step 4, the training of the classifier is performed only in case of absence of a classifier, or if the existing classifier does not match the required performance. The data resulting from the execution of the first six steps are free of noise and clustered by the batch, which allows direct analysis of further activities

without the need of additional pretreatment. As a visual aid, the summary of the algorithm and the transformation of the raw data in clean data is shown in **Figure 3.2**

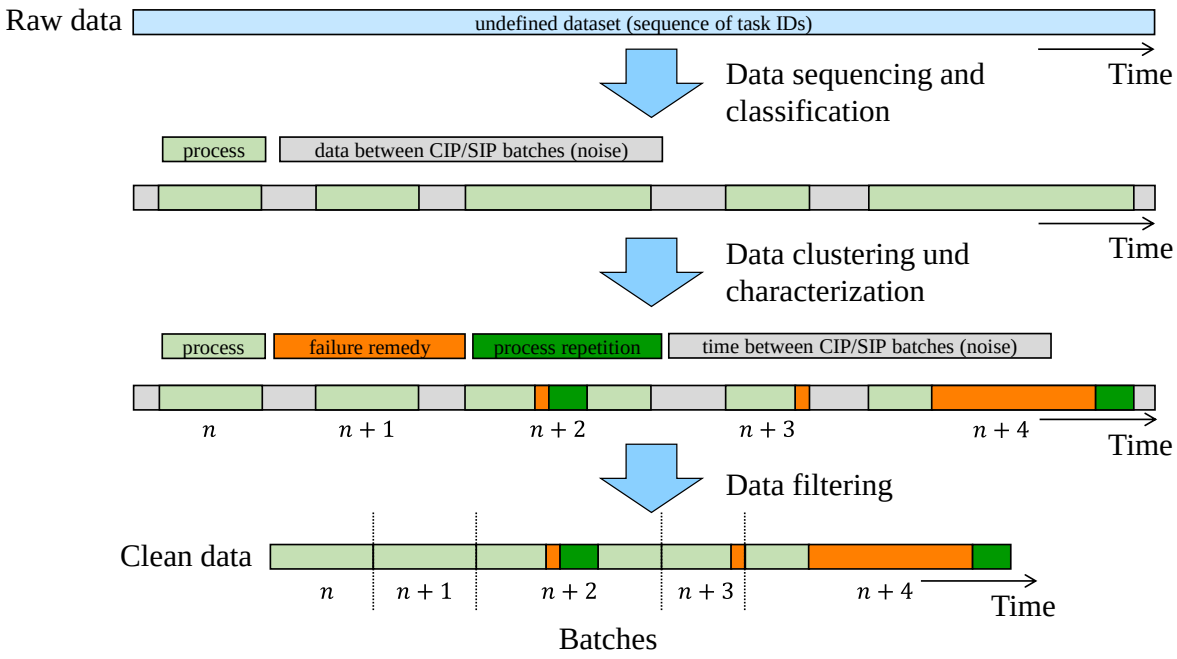


Figure 3.2 Summary of the algorithm applied to the raw data

3.2.1 Import raw data (step 1)

By choosing the time frame—i.e., the period for which the data to be preprocessed were recorded—and the resulting number of data points N_0 (with n as the counter), the data are selected from the manufacturing database $\boldsymbol{M}$. The step imports all the time stamps, with data containing the time of execution t_n, the task ID id_n and a categorical variable called "the Process-Noise (PN)" class, PN, for each data point. The PN class is manually evaluated to provide supervision, and is only used in the training of the classifier in step 4. The resulting dataset is referred to as "the raw dataset," $\boldsymbol{D}$.

3.2.2 Stem raw data (step 2)

In step 2, a stem process recipe is extrapolated, and the data from $\boldsymbol{D}$ is stemmed; step 2 is only performed if the raw data are a record of multiple similar processes—e.g., *pre*-CIP/SIP, *intra*-CIP/SIP or *post*-CIP/SIP, three similar processes found in the industrial case study. If the process recipes differ in a way that the stem recipe

cannot be identified, the stemming is skipped and the single recipes are used to preprocess the data independently. Analogously to natural language processing, where words are stemmed to a common root,[106] the stemming of the process recipes delivers a morphological root of the recipe, which is the common sequence of tasks among all processes (see **Figure 3.3**).

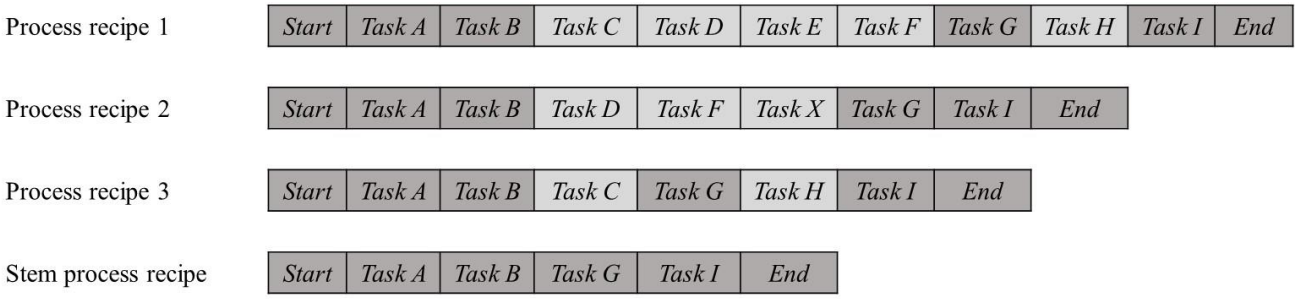


Figure 3.3 Practical example of the stemming procedure (step 2), where dark and light gray tasks are stem tasks moreover, non-stem tasks of the process recipe, respectively

Figure 3.3 shows a practical example of stemming three similar process recipes, where the tasks highlighted in light gray, namely *Task C*, *Task D*, *Task E*, *Task F*, *Task H*, and *Task X*, are not tasks in common and therefore are cut out. The stemming allows the processing of the entire dataset without the need of accounting for the presence of different process recipes, which would require applying the algorithm multiple times in parallel. After extrapolating the stem recipe, $\boldsymbol{D}$ undergoes the same treatment, where tasks not belonging to the stem recipe are temporarily eliminated, resulting in the stemmed dataset $\boldsymbol{D_1}$.

3.2.3 Sequence stemmed data (step 3)

The dataset $\boldsymbol{D_1}$ is sequenced in step 3; in this work, sequencing is referred as the procedure that quantifies the heterogeneity of a sequence of tasks $\boldsymbol{D_1}$ to the stem process recipe. The sequencing step consists of four main sub-steps: (3.1) find the Extremity Task Sequence (ETS), (3.2) calculate primer size, (3.3) calculate the Wagner–Fischer distance D_n^{WF} of the primer from the dataset, and (3.4) calculate the heterogeneity coefficient H_n (see **Figure 3.4**). For clarity, an explicative graphical representation of the entire sequencing algorithm is shown in **Figure 3.7** and the end of the section and in addition a step-by-step tutorial on the sequencing algorithm is shown in the appendix (See Appendix D.1 Sequencing tutorial)

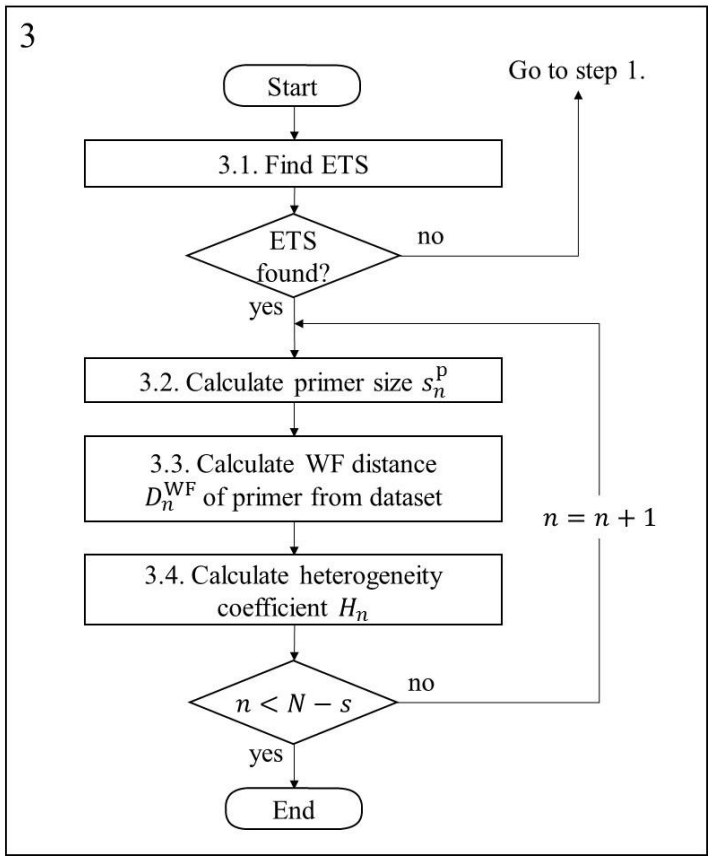


Figure 3.4 Diagram representing the sequencing step (step 3) of the algorithm

3.2.3.1 Find ETS (step 3.1)

In this step, $\boldsymbol{D_1}$ is screened to locate the position of the ETSs, p^{ETS_j}, where j is the counter. In a manner similar to DNA, where the start and stop codons are the extremities of a gene,[107] in this study, sETS and eETS are defined as the sequences of an s number of tasks located at the start and end of the process recipe, respectively. The position p^{ETS_j} of the ETS j is equal to the position p_n of a data point in $\boldsymbol{D_1}$ whenever $D_{n,n+s-1}^{\mathrm{ETS}}$ is equal to zero. The distance $D_{n,n+s-1}^{\mathrm{ETS}}$ is defined as the WF distance between the ETS—i.e., a fragment of the process recipe—and a sequence of tasks of the same size belonging to $\boldsymbol{D_1}$ starting from p_n. The distance $D_{n,n+s-1}^{\mathrm{ETS}}$ is quantified for every data point $n \in [1, N]$, with N being the size of the dataset $\boldsymbol{D_1}$.

After executing step 3.1, it is possible to continue to step 3.2 only if at least one ETS is found. The absence of an ETS suggests that $\boldsymbol{D_1}$ is too small and a full batch could not be identified in the selected time frame. In such a case, the algorithm asks to return to step 1, reselect a wider time interval and reimport a larger dataset from the database $\boldsymbol{M}$. Step 3.1 is concluded by collecting the execution times of the tasks relative to p^{ETS_j} into

the set T^{ETS} and by positioning s- and e-ETS on the dataset $\boldsymbol{D_1}$, as shown in the summary representation (see **Figure 3.7**)

3.2.3.2 Calculate primer size s_n^p (step 3.2)

The primer size s_n^{p} (not the ETS primer) is dynamically calculated at each point n of $\boldsymbol{D_1}$ following the algorithm shown in **Figure 3.5**. The dynamic calculation was adopted because it is known from the process that noise behaves differently according to its relative position to the ETSs—i.e., noise is mostly present between eETS and sETS (between batches), whereas operational disturbances are mostly present between sETS and eETS (within batches).

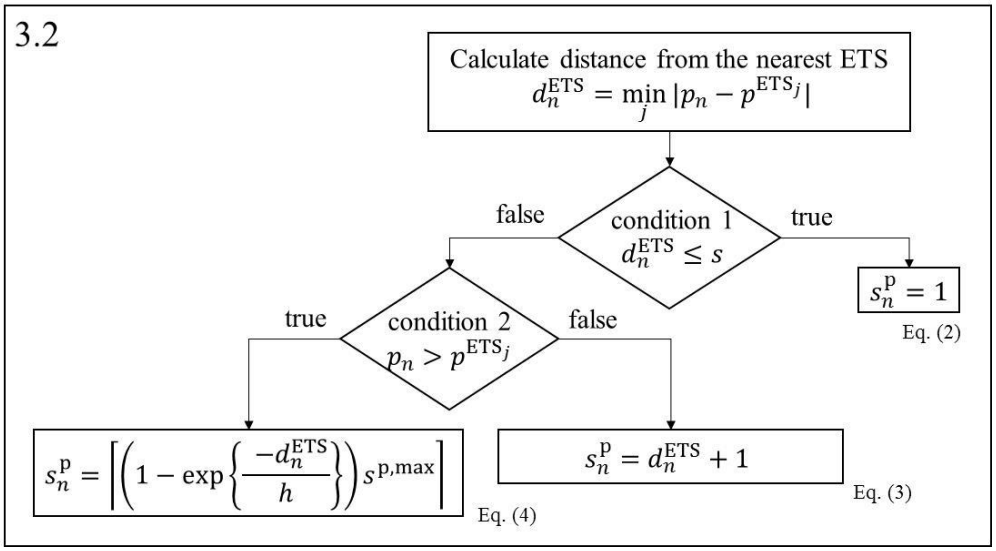


Figure 3.5 Decision tree diagram used for the dynamic calculation of the primer size s_n^p.

The distance of point n to the nearest ETS, d_n^{ETS}, is calculated using Eq. (3.1)

$$d_n^{\mathrm{ETS}} = \min_j |p_n - p^{\mathrm{ETS}_j}| \tag{3.1}$$

The primer size is calculated depending on the position of the point concerning the nearest ETS, according to Eqs. (3.2)–(3.4). The conditions highlighted in **Figure 3.4** differentiate between the various possible locations of the analyzed point, before, after or inside the ETS. By knowing that ETSs are the boundaries of a batch, the calculation of s_n^{p} can differentiate between positions as follows. If condition 1 is true, the point is positioned inside the ETS and Eq. (3.2) is used to calculate the primer size; else, condition 2 is tested.

$$s_n^{\mathrm{p}} = 1 \tag{3.2}$$

If condition 2 is true, the sequenced point is located before any ETS and Eq. (3.3) is used.

$$s_n^{\mathrm{p}} = d_n^{\mathrm{ETS}} + 1 \tag{3.3}$$

If condition 2 is false, the point is positioned after any ETS and therefore Eq. (3.4) is applied.

$$s_n^{\mathrm{p}} = \left\lceil \left(1 - \exp\left\{\frac{-d_n^{ETS}}{h}\right\}\right) s^{\mathrm{p,max}} \right\rceil \tag{3.4}$$

The parameters $s^{\mathrm{p,max}}$ and h are the maximal primer size and the rate of primer size change, respectively, and have to be specified by the user according to the case study, together with the parameter s. The selection of the primer size s_n^{p} is presented in **Figure 3.7**.

3.2.3.3 Calculate WF distance between the primer and dataset (step 3.3.)

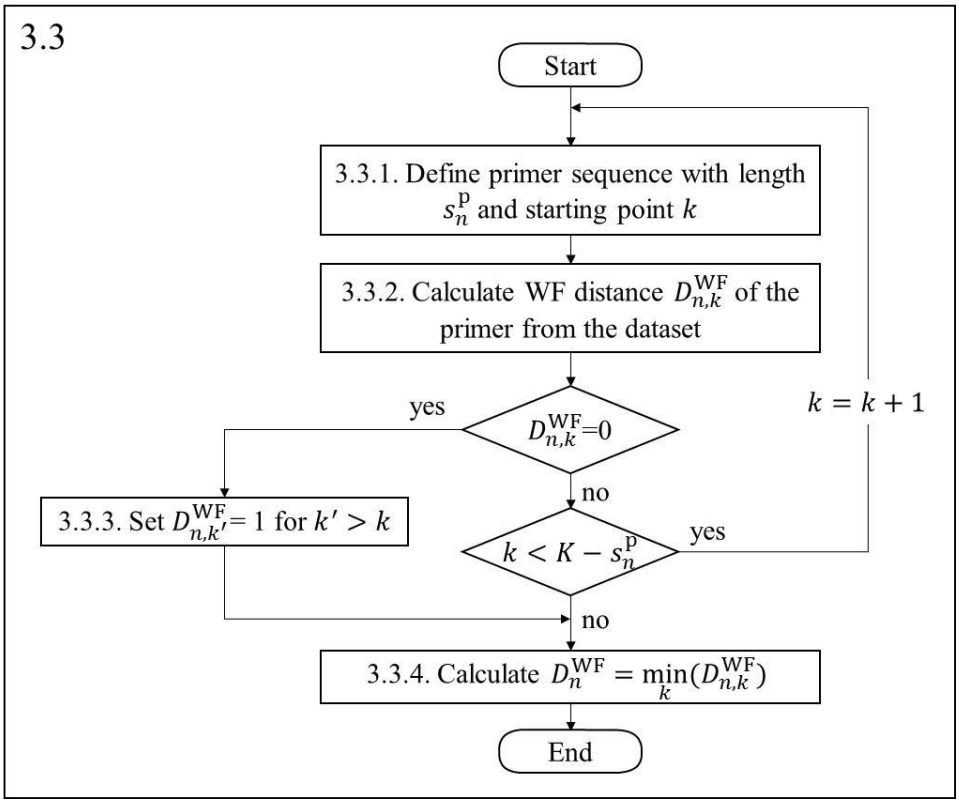


Figure 3.6 Algorithm for calculating the minimum distance D_n^{WF}.

The algorithmic procedure for calculating the minimal WF distance between the primer and the dataset $\boldsymbol{D_1}$ is shown in **Figure 3.6**. In step 3.3.1, the primer with size s_n^{p} at position p_n is defined as the fragment of the

process recipe from k to $k + s_n^{\mathrm{p}} - 1$ in a string vector—i.e., the vector components are in string form—; $k \in [1, K]$, is the counter of the tasks in the recipe and K is the total number of tasks in the recipe. In step 3.3.2, the WF distance $D_{n,k}^{\mathrm{WF}}$ between the primer and the data sequence is calculated at data point n for each k; if $D_{n,k}^{\mathrm{WF}} = 0$ before $k \geq K - s_n^{\mathrm{p}} + 1$, the distances $D_{n,k'}^{\mathrm{WF}}$ for $k' > k$ are set to 1 (step 3.3.3)—i.e., there is only one perfect match per sequence. In step 3.3.4, after calculating the $D_{n,k}^{\mathrm{WF}}$ for each primer k, the minimum distance D_n^{WF} is calculated using Eq. (3.5) and is defined as the WF distance of the data point n to the recipe.

$$D_n^{\mathrm{WF}} = \min_k D_{n,k}^{\mathrm{WF}}(s_n^{\mathrm{p}}) \tag{3.5}$$

3.2.3.4 Calculate heterogeneity coefficient H_n (step 3.4)

The last step of the sequencing algorithm is the calculation of the heterogeneity coefficient H_n (see **Figure 3.7**). The coefficient H_n quantifies the difference of the dataset fragment from the nearest primer from 1 to 0, where 1 stands for "different" and 0 for "identical", respectively. The heterogeneity coefficient H_n is defined as the WF distance D_n^{WF} normalized with the primer size s_n^{p}, as shown in Eq. (3.6).

$$H_n = \frac{D_n^{\mathrm{WF}}}{s_n^{\mathrm{p}}} \tag{3.6}$$

The way the process recipe is used in the algorithm to calculate the heterogeneity coefficient of each data point of $\boldsymbol{D_1}$ is shown in **Figure 3.7**.

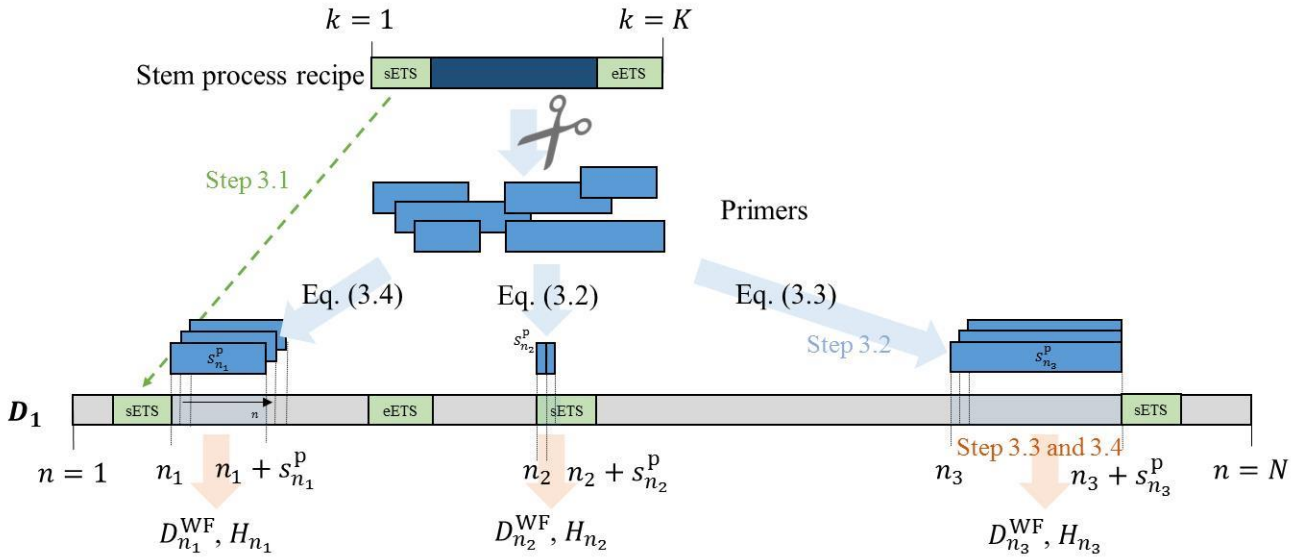


Figure 3.7 Summary of the sequencing step

3.2.4 Train classifier (step 4)

In this step, supervised machine learning is applied to classify data points into process data and noise. A supervised classification is possible because the noise is defined precisely and the raw data can be manually labeled. The noise constitutes a substantial part of the dataset therefore unsupervised outlier detection is not employable; also other unsupervised techniques, such as clustering, are not applicable because they cannot ensure the integrity of the data. The detailed algorithm used in this step integrates the results of the sequencing with the historical information on the classifier (**Figure 3.8**). The algorithm consists of three sub-steps: select a classifier (4.1), select predictor and response (4.2) and train the classifier (4.3).

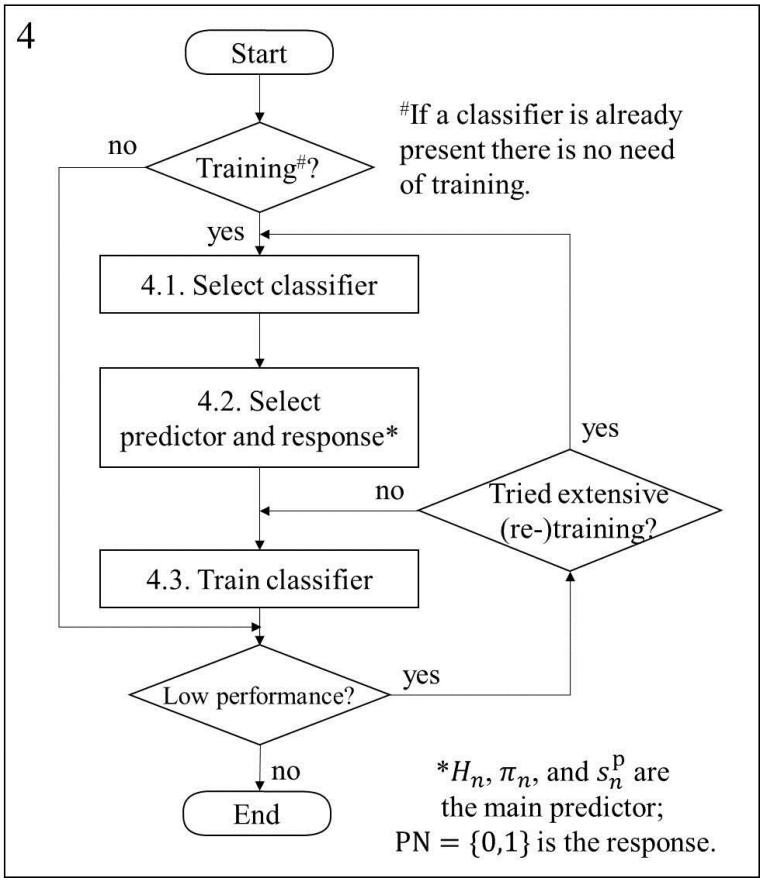


Figure 3.8 Algorithm for training the classification model.

3.2.4.1 Select classifier (step 4.1)

If no classifier is present, a classification model is selected in this step; examples of this are Decision Trees (DTs), support vector machines[108] and artificial neural networks[98]. Because of its application in an industrial framework, the appropriate model is selected by criteria such as accuracy, precision, training and evaluation velocity, as well as simplicity. Generally, a DT classifier is simple, fast and does not present any limitations regarding the size of the dataset or in dealing with nonlinearity; the DT is the first choice in the selection of the classifier. If the DT classifier shows poor performance, a further step in the algorithm allows the iteration of this step (4.1) and the selection of a different classifier model.

3.2.4.2 Select predictor (step 4.2)

In this step, predictors are defined to classify the noise; the heterogeneity coefficient H_n, the primer size s_n^p and the process coordinate π_n are selected as the initial/main predictors. The process coordinate of the primer, π_n, is defined as the starting position of the primer relative to the process recipe. The coordinate π_n is calculated

for each data point n of $\boldsymbol{D_1}$ and is defined as the counter k of the first task of the nearest primer—i.e., k: $\{k|\min_k D_{n,k}^{\mathrm{WF}}\}$—normalized with K (see Eq. (3.7)).

$$\pi_n = \frac{k}{K} \tag{3.7}$$

If during the utilization of the algorithm the quality of prediction does not match the expectation, additional predictors can be introduced to provide more information.

3.2.4.3 Train classifier (step 4.3)

The selected classifier is trained under supervision with a labeled training dataset, where the labels are PN classes, namely the response of the classification, $PN \in \{0,1\}$. The training strategies for achieving high-performance classifiers are selected, examples of which are the selection of the training dataset size, which must be large enough to be representative of the process, and the type of validation—i.e., cross-validation or independent validation dataset. Further discussions on the strategy selection can be found elsewhere.[98,105]

The model performance is evaluated by the F-score indicator, $Fscore$, which is the harmonic mean of precision, P_{class}, and recall, R_{class}, or the prediction accuracy, A_{class}. Each of these indicators is defined as in Eqs. (3.8)–(3.11), where $\boldsymbol{C}$ is the confusion matrix (see Appendix, section A.1) of the PN classes, and N_{obs} is the number of observations.[109]

$$R_{\mathrm{class}} = \frac{\boldsymbol{C}_{11}}{\boldsymbol{C}_{11} + \boldsymbol{C}_{10}} \tag{3.8}$$

$$P_{\mathrm{class}} = \frac{\boldsymbol{C}_{11}}{\boldsymbol{C}_{11} + \boldsymbol{C}_{01}} \tag{3.9}$$

$$Fscore = 2 \cdot \frac{1}{\frac{1}{R_{\mathrm{class}}} + \frac{1}{P_{\mathrm{pass}}}} \tag{3.10}$$

$$A_{\mathrm{class}} = \frac{\boldsymbol{C}_{11} + \boldsymbol{C}_{00}}{N_{\mathrm{obs}}} \tag{3.11}$$

Additional considerations on the performance can be made to improve prediction quality and stability over time. The number of false-positive ($\boldsymbol{C}_{10}$) and false-negative ($\boldsymbol{C}_{01}$) terms can be controlled by assigning different weights to the cost matrix—i.e., a training parameter—used in the training of the classifier. As mentioned above

(see Select classifier (step 4.1)) in the case of poor performance, either extensive training or the selection of a new classifier model is required.

3.2.5 Filter data (step 5)

The classifier is used to classify the data in the two PN classes. A value $PN \in \{0,1\}$ is assigned to each data point; the data points with $PN = 1$ are classified as "noise," and those with $PN = 0$ are classified as "process." Noise is removed from the dataset $\boldsymbol{D_1}$, resulting in the noise-free dataset $\boldsymbol{D_2}$with length $\widetilde{N}$.

3.2.6 Cluster data by batch (step 6)

The data that are free of noise is first clustered (step 6.2) with an unsupervised approach and subsequently reordered (step 6.3). The method of k-means clustering is applied iteratively to cluster the data points by batch. The sub-algorithm used for isolating single batches from $\boldsymbol{D_2}$ is shown in **Figure 3.9**.

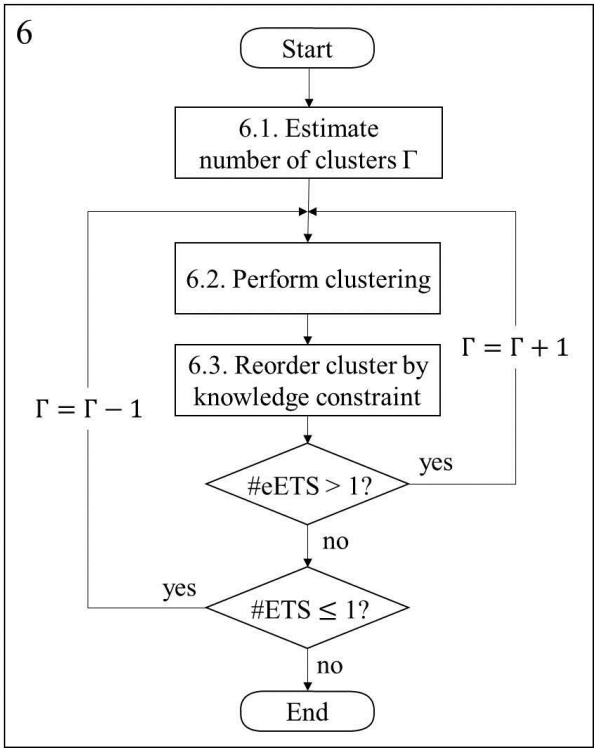


Figure 3.9 Algorithm used for clustering data points in batches.

The algorithm uses an iterative procedure to identify the number of batches executed in the time interval being analyzed.

First, the total number of clusters Γ—i.e., the number of batches—is estimated; as an indication, the number of sETS is an appropriate first estimation. Second, the k-means clustering algorithm presented by Macqueen (1967) is applied;[110] the two-dimensional k-means clustering mechanism uses the execution time t_n of each task—i.e., of each data point n —as one of the dimensions. The other dimension used in the clustering is the normalized position of the task in the recipe, namely the process coordinate π_n. The clusters are defined as sets C_i, with $i \in [1, \Gamma]$ containing the data points from $\boldsymbol{D_2}$ as the result. Third, a reordering algorithm (see **Table 3.1**), imposes on each cluster C_i, except for the first one, C_1, an sETS data point at its lowest extremity; the algorithm enforces a clear split between batches in a supervised way. The first cluster is excluded because the starting time of the imported raw dataset does not necessarily coincide with an sETS. Because the operational disturbance is still present in the data set, not every sETS implies the beginning of a commercial batch. The semi supervised clustering approach is necessary because failure can cause the repetition of the whole process, including the setting of sETS.

Table 3.1 Algorithm used to reorder the clusters into batches.

define $t_{u,i}$ where u is the position of the data point inside the cluster i (containing U number of points) for each $n \in C_i$
  for each $i > 1$ and while $t_{1,i} \notin T^{\mathrm{ETS}}$
    set $t_{U,i-1} = t_{1,i}$
    set $t_{1,i} = t_{2,i}$
    for each $u > 1$, $t_{u,i} = t_{u-1,i}$
    return
  return
return C_i

Each cluster is analyzed to identify the number of batches, and if a cluster contains more than one eETS, the algorithm will suggest increasing the initial Γ; two different batches might be assigned to a single cluster. Likewise, if a cluster does not contain any, one batch is probably split into two clusters, and the algorithm will suggest decreasing the initial Γ. In both cases, the algorithm is iterated from step 6.1 until the conditions are satisfied.

After concluding the clustering, the time extrema—i.e., $t_{1,i}$ and $t_{U,i}$—of each cluster i determine the start and end of each batch. These boundaries are used to re-gather the points, which were removed by the stemming, belonging to each batch from the dataset $\boldsymbol{D}$.

3.2.7 Characterize batch (step 7)

After clustering, each batch is characterized by identifying the process type—e.g., *pre-*, *intra-* and *post-*CIP/SIP—through a string search. Subsequently, the data points related to operational uncertainty are identified and labeled as *repetition*, *remedy* and *alarm* points, whereas normally executed tasks are labeled as *normal*. The time lost because of operational uncertainty is quantified and presented as an additional result of the step.

3.3 Preliminary study

The presented algorithm was applied to preprocess the raw manufacturing records of an industrial CIP/SIP process. Speed, prediction quality and stability over time are essential factors in developing an algorithm applicable to the industry. Hence, the effects of various parameters on these factors were analyzed as a preliminary study aimed at an efficient design of the algorithm.

3.3.1 Speed and prediction quality

In the first study, the effect of the two sequencing parameters, s and $s^{\mathrm{p,max}}$, on the computational time for sequencing and prediction quality was analyzed in a multiobjective evaluation. Long computational times were foreseen for calculating the distance between the (relatively large) strings. The computational time complexity of the approximated string-matching algorithm was reported using the computational order, $O(m \cdot m')$, [103,111], where m and m' were the lengths of the two strings compared. In fact, after a preliminary testing phase, it was confirmed that the sequencing algorithm, and, in particular, the calculation of WF distances, was the speed bottleneck for the entire algorithm. For industrial use, however, a short computational time is preferable, especially if the algorithm is used to preprocess big datasets (~60,000 data points in one year of data).

The quality of the prediction was evaluated by the F-score, which in turn was analyzed by changing the combination of the sequencing parameters to determine the most suitable design. From the training dataset,

fragments of 3,000 data points were randomly selected (10,000 iterations), and their predictors were used to train and validate the DT classifier (10-fold cross-validation).

The results of the previous preliminary analyses were combined in a multiobjective evaluation, where a low computational time and high F-scores were desired; the cumulative cost presented in Eq. (3.12) was used for the evaluation.

$$Cumulative\ cost = computational\ \ time' + \left(\frac{1}{Fscore}\right)' \tag{3.12}$$

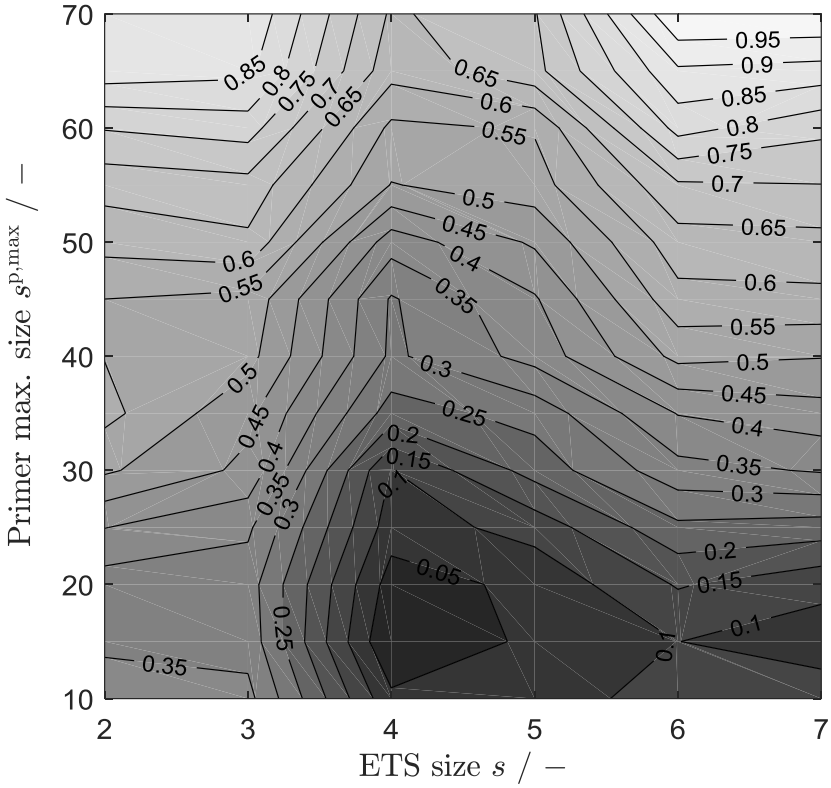


Figure 3.10 Multiobjective evaluation of ETS and maximum primer sizes.

The result of the multiobjective evaluation (see **Figure 3.10**) presents the scaled cumulative cost; the minimum of this cost was found with the values s = 4 and $s^{\mathrm{p,max}}$ = 20. In addition, a simple sensitivity analysis that considered the response and computational efforts found that the value of parameter h was 25.

3.3.2 Stability over time

Continuous process improvement in manufacturing is usually accompanied by process modification, which sometimes leads to a change in the behavior of the data recorded during production. In the pharmaceutical

industry, the performance of manufacturing facilities is continuously improved for optimality.[104,112] In this concern, a control strategy—i.e., retraining strategy—of the algorithm performance is needed, requiring a minimal human intervention—e.g., labeling data.

The stability over time of the prediction performance was evaluated, and the result is shown in **Figure 3.11**.

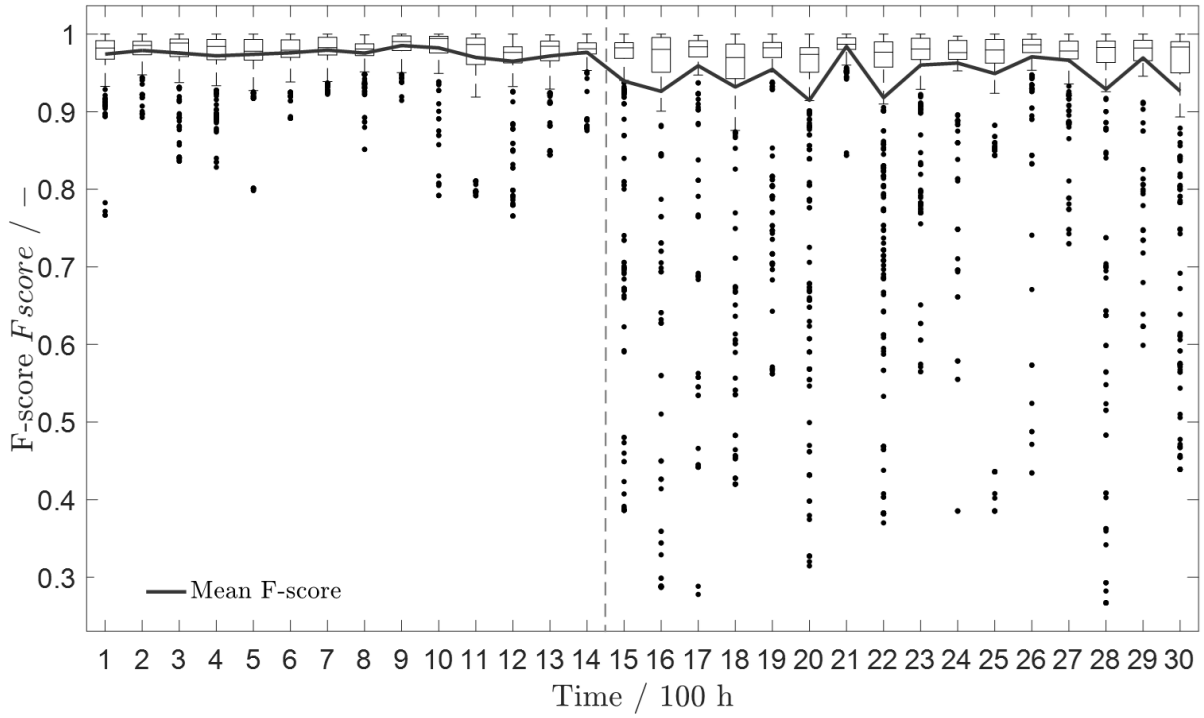


Figure 3.11 Analysis of the stability of the predicted quality over time.

It can be observed that after 1,500 h, the predicted performance decreased, suggesting the need for retraining. Therefore, to guarantee time-stability and high-performance, the interval for retraining the model using labeled records from recent batches was set at 1,500 h. The reason of approximately 3 months of model stability could be attributed to the continuous manipulation on the plant during the setup and format change operations; it was hypothesized that the manipulation of could have induced trends in the data over, which resulted in the divergence of the predictor from the training dataset. Additional analyses would be required to investigate the real reason of the drop in stability of the model, or process drift; however, because it is not the main scope of the thesis, the analysis is not shown in this work.

3.4 Application of the algorithm to commercial data

The algorithm was applied to three sets of data, each representing a 2-month record from a commercial CIP/SIP process. No retraining was required within each dataset because the timespan of the raw data was shorter than 1,500 h.

3.4.1 Import raw data (step 1)

The manufacturing data was imported, evaluated, classified, and categorized. The imported datasets are listed in **Table 3.2**.

Table 3.2 Import information of the datasets

Dataset ID	N_o	Timespan [h]	Execution period
$\boldsymbol{D}$**-1**	10,018	1,415	March–April
$\boldsymbol{D}$**-2**	8,696	1,462	June–July
$\boldsymbol{D}$**-3**	10,918	1,451	October–November

3.4.2 Stem raw data (step 2)

A stem recipe was created from the *pre-*, *intra-* and *post*-CIP/SIP recipes using the set logic operations shown by Eq. (3.13):

$$\begin{cases} R_{\text{stem}} \subseteq R_{\text{pre}} \quad (252 \text{ tasks}) \\ R_{\text{stem}} \subseteq R_{\text{intra}} \quad (186 \text{ tasks}) \\ R_{\text{stem}} \subseteq R_{\text{post}} \quad (180 \text{ tasks}) \\ R_{\text{stem}} = R_{\text{pre}} \cap R_{\text{intra}} \cap R_{\text{post}} \\ R_{\text{elim}} = R_{\text{stem}} \ \triangle \ (R_{\text{pre}} \cup R_{\text{intra}} \cup R_{\text{post}}) \end{cases} \tag{3.131}$$

R_{stem} is the set containing the task IDs of the stem recipe. The sets R_{pre}, R_{intra} and R_{post} contain the task IDs of the recipes of the *pre-*, *intra-* and *post*-CIP/SIP processes, respectively; R_{elim} is the set containing the IDs of the tasks eliminated during the stemming process.

The sets R_{stem} and R_{elim} contained 165 and 108 task IDs, respectively. Data points that their id_n were contained in R_{elim}, were eliminated from the raw datasets $\boldsymbol{D}$**-1**, $\boldsymbol{D}$**-2** and $\boldsymbol{D}$**-3**, resulting in the datasets $\boldsymbol{D_1}$**-1**, $\boldsymbol{D_1}$**-2** and $\boldsymbol{D_1}$**-3** with sizes of 8,254, 6,969 and 8,762 data points, respectively.

3.4.3 Sequence stemmed data (step 3)

The data was sequenced using the optimal design found in the preliminary study (see section Preliminary study). The numbers of sETSs and eETSs of these sets are presented in **Table 3.3**.

Table 3.3 Number of the identified sETS and eETS.

Dataset ID	Number of sETS	Number of eETS
$\boldsymbol{D_1}$**-1**	43	35
$\boldsymbol{D_1}$**-2**	38	29
$\boldsymbol{D_1}$**-3**	43	35

From **Table 3.3**, it is notable that the number of sETSs and eETS in each dataset is different. The difference was attributed to operational uncertainty, failures in and repetitions of tasks, which resulted in fewer eETS than sETS.

The datasets were sequenced and the heterogeneity coefficients H_n were calculated; as a result of the sequencing, the tuples of the predictors s_n^{p}, H_n and π_n could be obtained for each point n of the dataset.

3.4.4 Train classifier (step 4)

The DT classifiers were trained, and the training datasets and subsets of the three datasets were used to train the model (maximum number of splits = 10, cost matrix $\begin{bmatrix} 0 & 1 \\ 10 & 0 \end{bmatrix}$). The training datasets were manually classified by defining the PN class for each data point. Ten-fold cross-validation was used to validate the classifier, and the F-score was used to characterize its performance. A summary of the training activity performed in this step is given in **Table 3.4**.

Table 3.4 Summary of the activity in step 4.

Dataset ID	$\boldsymbol{N}$	**Training subset size**	**F-score**	**Error**
$\boldsymbol{D_1}$**-1**	8,254	3,000	0.9910	0.0153
$\boldsymbol{D_1}$**-2**	6,969	3,000	0.9911	0.0143
$\boldsymbol{D_1}$**-3**	8,762	3,000	0.9952	0.0063

Table 3.4 shows that the classifier was validated, and exhibited very high performance in each case.

3.4.5 Filter data (step 5)

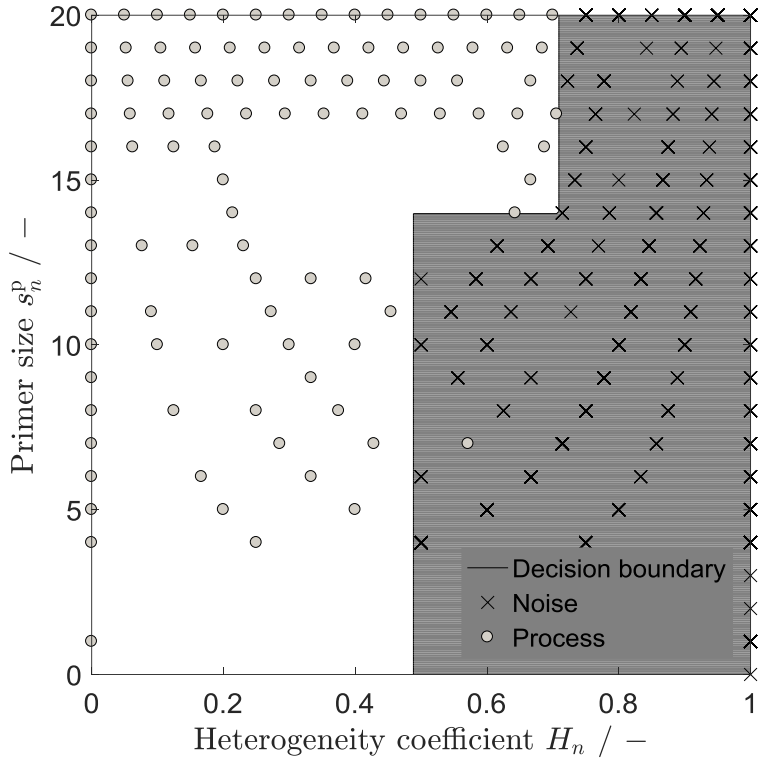


Figure 3.12 The result of the classification; data classified as "noise" (crosses) and as "process" (circles) are divided by the decision boundary.

The classification results of the dataset $\boldsymbol{D_1}$**-1**, of size N, in which each data point is plotted as the combination of s_n^{p} and H_n are shown in **Figure 3.12**. The results for the other datasets, $\boldsymbol{D_1}$**-2** and $\boldsymbol{D_1}$**-3**, are found in the appendix (see **Figure A.1**). The classifier delivered a net classification, i.e., a clean decision boundary that allowed deleting the data points classified as noise from the datasets, and shrinking in their sizes to $\widetilde{N}$, yielding the noise-free datasets $\boldsymbol{D_2}$**-1**, $\boldsymbol{D_2}$**-2** and $\boldsymbol{D_2}$**-3**. The result of the classification is shown in **Table 3.5**, which lists the percentage of shrinking from the raw datasets ***D*****-1**, ***D*****-2** and ***D*****-3**.

Table 3.5 Summary of the result of step 5

Dataset ID	$\widetilde{N}$	Percentage of shrinking
$\boldsymbol{D_2}$**-1**	6,775	32%
$\boldsymbol{D_2}$**-2**	5,456	37%
$\boldsymbol{D_2}$**-3**	6,596	39%

From the percentage of shrinkage, it is notable that 30–40% of the data was not essential; hence, a large part of the data could potentially jeopardize the success of the preprocessing.

3.4.6 Cluster data by batch (step 6)

On the shop floor, the commercial CIP/SIP batches are executed at different times and are usually separated by sterile filling or time-consuming setup operations. Thus, time was used as the first clustering dimension, and π_n as the second dimension. The results of the clustering on the datasets $\boldsymbol{D_2}$**-1**, $\boldsymbol{D_2}$**-2** and $\boldsymbol{D_2}$**-3** are shown in **Figure 3.13** (see section Cluster data by batch (step 6)). In this figure, each gray area represents one cluster, i.e., one batch. In **Figure 3.13**, the whole raw dataset ***D*-1** was plotted to give a complete image of the outcome of the first six steps of the algorithm (for sets ***D*-2** and ***D*-3**, see **Figure A.2** in the appendix).

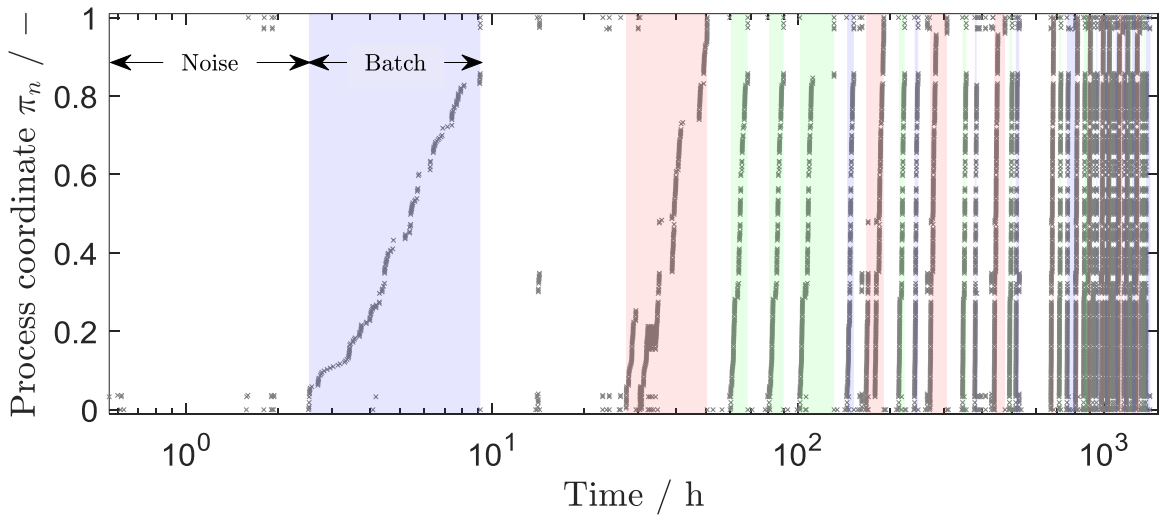


Figure 3.13 The result of clustering on dataset ***D*-1**.

3.4.7 Characterize batch (step 7)

After isolating the batches, the process type of each batch was identified by recognizing specific tasks using set logic; the algorithm is presented in **Table 3.6**. The results are shown in **Figure 3.13**, where *pre-*, *intra-* and *post-*CIP/SIP are highlighted in red, green and blue, respectively.

Table 3.6 Algorithm used to characterize the process type.

If $\exists n \in C_i \mid id_n \in \{R_{pre} \setminus (R_{intra} \cup R_{post})\}$;
C_i *is pre-CIP/SIP*
If else $\exists n \in C_i \mid id_n \in \{R_{intra} \setminus (R_{pre} \cup R_{post})\}$;
C_i *is intra-CIP/SIP*
If else $\exists n \in C_i \mid id_n \in \{R_{post} \setminus (R_{pre} \cup R_{intra})\}$;
C_i *is post-CIP/SIP*
end

In addition to identifying the process type, the data conditioned by operational uncertainty was labeled as *repetition*, *remedy* or *alarm*. Whenever a failure occurs in the execution of a task, the process stops, and an alarm is automatically turned on—i.e., an alarm task is executed—; the alarm stays "on" until the remedy operations start. All alarm tasks, such as "Stop during decontamination," "Stop during rising" or "Stop during drying" are listed on the alarm list. Because of GMP, remedy operations are always preceded by an alarm, and in the case of failure, the set of predefined tasks that preceded the failing tasks is repeated. In summary, because of GMP, the actions to be taken in case of failure are known and predefined; hence, the search of alarms in the dataset was done through conditional reasoning (the algorithm is shown in the appendix, section A.3).

The output of this step and the overall output of the algorithm, namely the clean data, is a structured dataset divided into batches; for each batch, the process type is indicated, and the tasks are labeled.

3.5 Results of the implementation

The algorithm could be implemented with historical data from 6 months of production. As a validation of the algorithm, the result was compared with the outcome of the same evaluation that was manually performed. The manual evaluation of 29,632 data points took approximately 24 h, whereas the automated evaluation of the same dataset (***D*-1**, ***D*-2** and ***D*-3**) took 66 min (–86.2%), 56 min (–88.3%) and 140 min (–70.8%), respectively, with Intel® Core i5 2.3GHz, 8 RAM. The algorithm performance was measured with three indicators, namely the total yield, the batch deviation, and third, the misclassification; the indicators compared the outcomes of the automated and the manual pre-processing, the latter being the current practice. The first is defined as defined as

the ratio of number of batches identified by the algorithm and by the current practice; the second is defined as the time deviation between the duration of a batch recognized by the algorithm in the real duration of a batch. The last indicator is defined as the ratio of misclassified data points and the total number of the evaluated points. **Table 3.7** shows a summary of the algorithm performance indicators for the three datasets. The detailed performance analysis is shown in **Tables**. **A.1–A.3** in the appendix.

Table 3.7 Results of the performance of the algorithm.

Dataset ID	Yield	Mean batch deviation	Median batch deviation	Misclassification rate
***D*-1**	97%	29±119%	0.03%	4.1%
***D*-2**	94%	5.8±16%	0.02%	1.1%
***D*-3**	95%	1.3±4.6%	0.03%	1.5%

Table 3.7 shows high yields in general, indicating that the algorithm is capable of recognizing single batches inside a continuously measured dataset. The mean batch deviation and the misclassification rate appear to be relatively high for ***D*-1**, whereas the median batch deviation is comparable with the other datasets. Analyzing the single performance of each batch in ***D*-1**, it is notable that 10% of the batches show an extremely high deviation. The special cause for this high deviation was the plant maintenance performed in January and February. Thus, it can be concluded that the algorithm is generally valid, but requires more attention when dealing with records generated by a still unstable process, as it is the case shortly after the maintenance.

3.6 Novelties and limitations of the algorithm

Our approach presented mainly four novelties: First, the analogy between the time series of batches and the DNA strain allowed the integration of well-established and robust approaches, such as approximate string matching in the supervised removal of noise from manufacturing records. Second, the definition of two types of data disturbance, operational disturbance, and noise, differentiates the disturbance caused by the operations and the noise that is dependent on the continuous data recording system. Such differentiation supports recognizing and isolating data that originated from operational uncertainty in the process. Third, the adaptive clustering algorithm, in step 6, recognizes single batches without the need of the number of batches contained in the dataset as input; such information is usually required for k-means clustering. Finally, compared with tools

used in database management and data retrieval, such as structured query language, the presented algorithm can deal with uncertainty to robustly isolate processes from continuously measured datasets.

The novel algorithm also introduced various improvements compared with manual preprocessing. The manual procedure required a long time to preprocess the data and such a time investment could render immediate retrospective analysis and even process monitoring prohibitive. By contrast, the relatively short processing time presented in this work showed the potential applicability of the automated preprocess algorithm on the shop floor. The introduction of automation not only reduces the risk of corrupting the integrity of sensitive data but also possibly facilitates the fast retrieval of data during reviews and inspections by regulatory agencies such as United States Food and Drug Administration (FDA), as mentioned by Meneghetti et al. (2016).[105]

Limitations could be found in the use of the algorithm in day-to-day operations. First, the execution time required in the sequencing step is still long, and this limits the speed of providing an online result; second, the rapidity of the preprocessing is further reduced if the stemming step is not possible because of the presence of multiple non-similar recipes, resulting in the iteration of the algorithm. Third, the algorithm required a large amount of data to deliver meaningful results. Because of such limitations, at state of the art, the algorithm can only be applied for process monitoring and retrospective analyses.

3.7 Alternative utilization of the clean data in the context of operations support

The outcome of the algorithm, the clean data, finds potential application in the current trending and monitoring practices, as well as in support of decision-making in process improvement. Root Cause Analysis (RCA) is a commonly used method performed by technicians in the industry whenever process failures occur, and their outcomes are recorded in a database according to cGMP guidelines. As an alternative utilization, the clean data were incorporated into the RCA to create a quantitative RCA. Two Ishikawa representations of the process failure for the previous 6 months of operations based on the 6M-model [113] are shown in **Figure 3.14**.

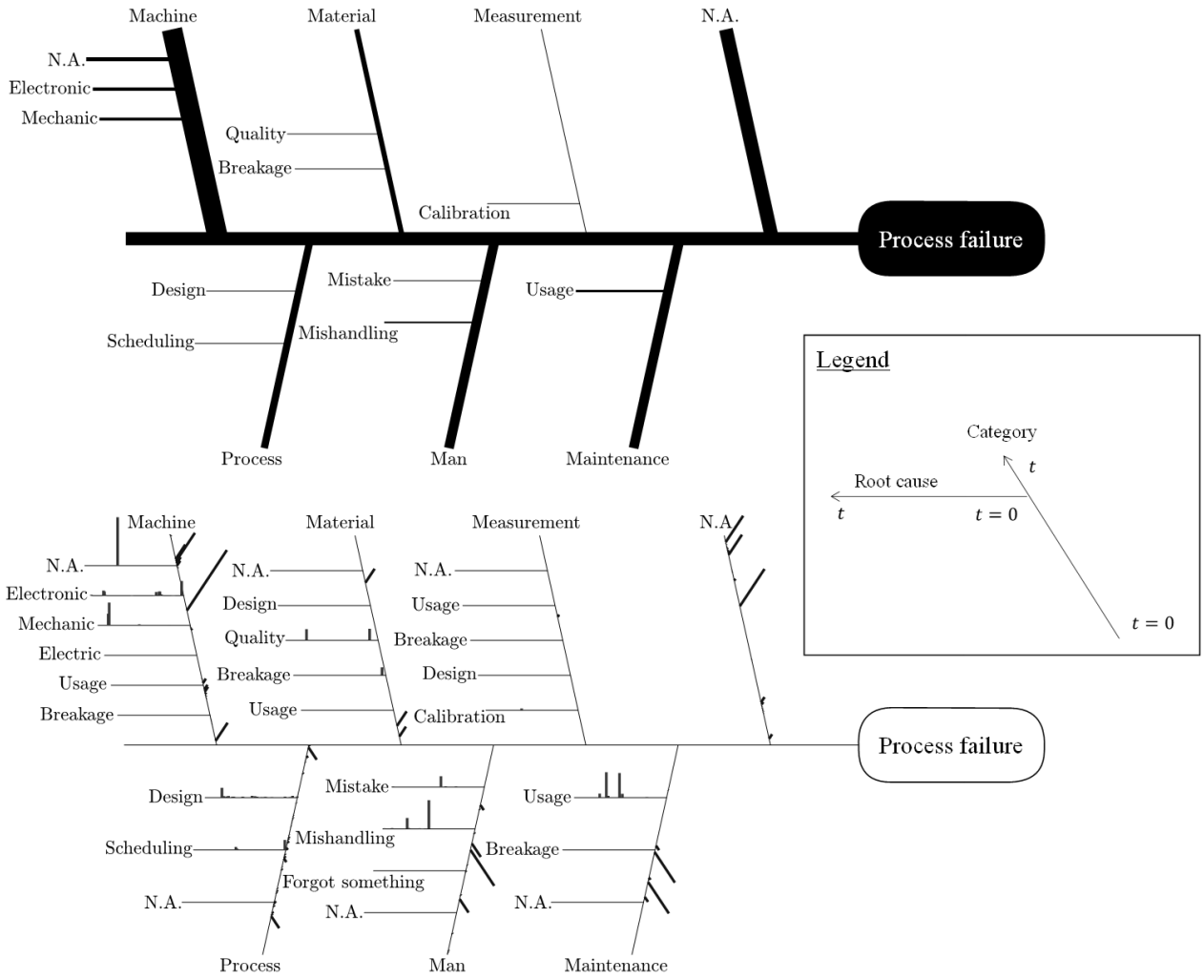


Figure 3.14 Cumulative (top) and evolutionary (bottom) fishbone diagrams.

In the cumulative fishbone diagram (**Figure 3.14**), the thickness of the "bones" represents the cumulative impact of the *category*—i.e., machine, material, measurement, maintenance, man, and process—and the *root cause*—e.g., electronic, design or breakage—of the failures. Such a representation facilitates the fast recognition of important causes of failures. The evolutionary diagram (see **Figure 3.14**, bottom) presents a time evolution of the causes of failures. Each "bone," main and secondary, of the diagram represents a time axis (from $t = 0$ to t), which evolves from the stem to the tip of the "bone". The implementation of this RCA tool can standardize the outcome of the analysis by removing subjective factors in the analysis, such as naming inconsistencies—e.g., "man," "human" and "operator" as synonyms. The automated integration in the RCA and the impact quantification provides a simple measure to quantify priorities in decision-making.

3.8 Conclusion

In this chapter, an algorithmic approach for the preprocessing of manufacturing records in biopharmaceutical manufacturing while considering operational uncertainty was presented. The algorithm can automate the identification of batches within a continuously measured historical record and the translation of the record in the form of timestamps—i.e., tuples of natural language and time information—into clean data. DNA-like sequencing techniques, such as the approximate string-matching algorithm, were used to transform natural language data into quantitative data. Such data were subsequently used for the classification of noise by supervised machine learning, with the use of DT models. After clustering the noise-free dataset into single batches, the impact of operational uncertainty was characterized. The outcome of the characterization was the isolation of the time invested in operations that are not expected in the normal operational framework but could be attributed to the presence of failures in the process. Additionally, two novel graphical representation methods, namely the cumulative and evolutionary Ishikawa diagrams, have been developed to facilitate a rapid and structured way to integrate process data and practitioner analyses in decision-making.

In the preliminary study, the algorithm's precision, speed, and stability over time were assessed for selecting the best combination of parameters to be applied in the case study. Then, the algorithm was demonstrated using three 2-month historical datasets of a CIP/SIP process in a sterile biopharmaceutics DP manufacturing factory. The algorithm reduced the time down to 11.7% compared with manual data preprocessing, while maintaining high data quality and integrity. The clean data were then exploited in the newly developed RCA with yielding quantitative and chronological insights for improving the operations.

The algorithm presents some limitations regarding the processing speed and minimum data amount requirement to be applicable to real-time operations on the shop floor. However, the capability to generate clean data in an automated manner opens opportunities in the analysis of process data. The presented work is a preliminary study on the digitalization of biopharmaceutical manufacturing process and is necessary for the introduction of Industry 4.0 and Smart Manufacturing concepts in this industry (see **Chapter 5**). The output of the algorithm is used in **Chapter 4** and **Chapter 5** as an input for the performance assessment and the failure prediction.

3.9 Nomenclature

A_{class}	Accuracy of the DT classifier	–
$\boldsymbol{C}$	Confusion matrix	
C_i	Cluster i	
d_n^{ETS}	Distance of data point n from the nearest ETS	–
$\boldsymbol{D}$	Raw dataset of size N_o	
$\boldsymbol{D_1}$	Stemmed dataset of size N	
$\boldsymbol{D_2}$	Stemmed and noise-free dataset of size $\widetilde{N}$	
D_n^{WF}	WF distance at between primer and dataset at data point n	–
$D_{n,n+s-1}^{\text{ETS}}$	WF distance at between ETS primer and dataset at data point n	–
$D_{n,k}^{\text{WF}}$	WF distance at between primer and dataset at data point n	–
$Fscore$	F-score of the DT classifier	–
Γ	Total number of clusters	–
h	Change velocity factor of the primer size	–
H_n	Heterogeneity coefficient of data point n	–
id_n	Task ID of data point n	
k	Position in the process recipe	
K	Total number of tasks in the process recipe	–
$\boldsymbol{M}$	Manufacturing database	
m	General length of a string	–
m'	General length of a string	–
n	Counter of data points used in the dataset	–
N	Size of the stemmed dataset	–
N_o	Size of the raw dataset	–
$\widetilde{N}$	Size of the stemmed and noise free dataset	–
N_{obs}	Size of the dataset for the validation of the DT classifier	–

O	Computational order of the approximate string matching algorithm	
p^{ETS_j}	Position of the ETS j with respect to the dataset	
p_n	Position of data point n	
π_n	Process coordinate for data point n	–
P_{class}	Precision of the DT classifier	–
PN	Process-Noise class	
R_{class}	Recall of the DT classifier	–
R_{stem}	Set containing the task IDs of the stemmed process recipe	
R_{pre}	Set containing the task IDs of the process recipe for pre-CIP/SIP	
R_{intra}	Set containing the task IDs of the process recipe for intra-CIP/SIP	
R_{post}	Set containing the task IDs of the process recipe for post-CIP/SIP	
R_{elim}	Set containing the task IDs not contained in the stemmed process recipe	
s	Size of the ETS primer	–
$\mathrm{s}_n^{\mathrm{p}}$	Size of the primer for data point n	–
$s^{\mathrm{p,max}}$	Maximum primer size	–
t_n	Execution time of data point n	h
$t_{u,i}$	Execution time of data point u in cluster i	h
T^{ETS}	Set containing the ETS execution times	
U	Total number of points contained in each cluster C_i	–
u	Counter of data points inside each cluster C_i	

Chapter 4: Uncertainty-conscious methodology for process performance assessment in biopharmaceutical drug product manufacturing

(*Based on* " Casola G, Sugiyama H, Siegmund C, Mattern M. Uncertainty-conscious methodology for process performance assessment in biopharmaceutical drug product manufacturing. *AIChE J*. 2018;64:1272-1284"

4.1 Introduction

As mentioned in the introduction, characteristics, as can be seen in CIP/SIP such as semiautomation, time intensity, and the need to follow GMP, can all make process improvement challenging. For CIP/SIP, both mechanical issues and operator behavior could lead to success or failure of the process, and variability in the process regardless of its success or failure. Both nonroutine events producing process failure and process variability can lead to operational uncertainty and affect process performance. Another important characteristic that needs to be considered is GMP. The FDA (2016) and other national authorities supervise manufacturing processes to guarantee that pharmaceutical products are of high quality.[90] In modifying commercial processes, GMP requires time- and resource-intensive revalidation and requalification procedures,[19] which hinder substantial changes in the process even when the change itself is beneficial. In the case of CIP/SIP, modifications of the task sequences would have to go through the validation procedures independently of the increase in process performance. Operational uncertainty and GMP, therefore, need to be considered when seeking to improve process performance in the manufacturing of biopharmaceutical DPs.

Generally, approaches such as Lean Manufacturing and Lean Six Sigma (LSS) are pillars for the continuous improvement of process performance and have been applied in the pharmaceutical industry.[26,114] LSS is a collection of managerial and statistical methods such as Shewhart's control charts, root cause analysis, and value stream mapping that are applied to reduce variability and waste in manufacturing processes.[115,116] For example, Dassau et al. (2006) combined LSS with design and control approaches for yield enhancement in biopharmaceutical DS manufacturing,[117] and Boltic et al. (2016) showed the effectiveness of LSS in improving quality assurance in small-molecule DS manufacturing.[112] These studies applied well-known industrial approaches for process performance improvement but lacked a rigorous description of the complex pharmaceutical environment in which these processes occurred.

Besides LSS, modeling, simulation, and optimization, which have always been the core of (PSE), are applied intensively in drug manufacturing.[54,118,119] Reklaitis et al. described the challenges faced by the industry in process development and optimization as an opportunity for PSE research.[120] Sundaramoorthy et al. (2012) developed an uncertainty-conscious framework for capacity optimization in a pharmaceutical supply chain;[121] Costa (2015) solved a scheduling problem in pharmaceutical batch processes by means of hybrid genetic

optimization;[122] and Singh et al. (2014) proposed a hybrid control system combining model predictive control with proportional–integral–derivative control for continuous tablet manufacturing.[60] In recent years, researchers have intensively focused on the deterministic description and modeling of drug manufacturing processes, more specifically for small molecules. Jolliffe and Gerogiorgis (2015 and 2016) presented economic and environmental feasibility studies of continuous manufacturing processes of ibuprofen and artemisinin.[123,124] Rogers and Ierapetritou (2014) reported the potential for the industry to apply deterministic modeling and optimization through a case study in DP manufacturing.[74] Boukouvala et al. (2012) proposed an approach that applies flowsheet modeling and sensitivity analysis to analyze the efficiency and robustness of DP manufacturing.[59]

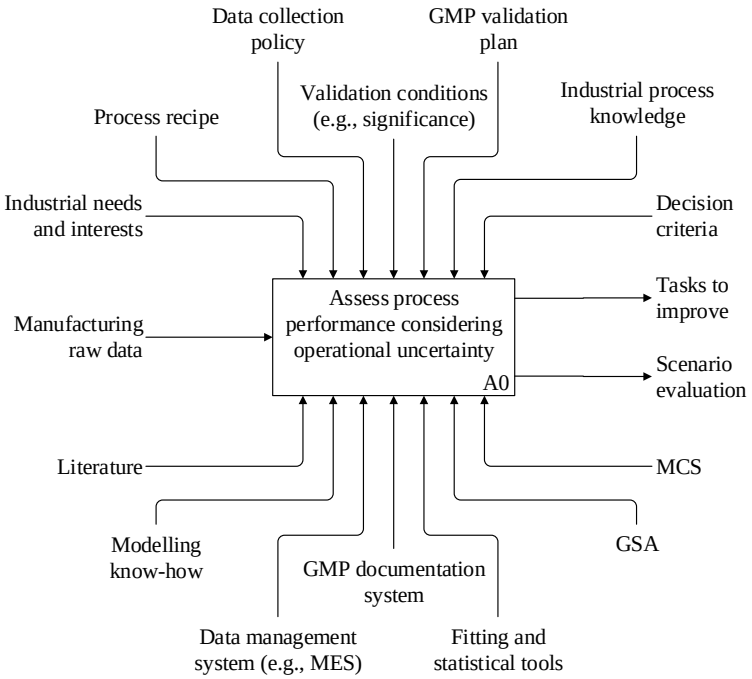
Currently, sensitivity and uncertainty are core topics in the field of PSE. Among other sensitivity analysis techniques, Global Sensitivity Analysis (GSA)—e.g., Sobol's index sensitivity analysis—is a broadly used technique for identifying factors that influence process performance, and for the characterization of uncertainty.[125] Bahakim et al. (2018) studied the optimal design of large-scale chemical processes under uncertainty using a ranking-based approach.[126] Cadini and Gioletta (2016) proposed an algorithm based on stochastic simulation for estimating failure probabilities of systems affected by uncertainty and applied the algorithm to two case studies.[127] In two review papers, Grossman et al. (2005 and 2016) emphasized the importance of enterprise-wide optimization and the advances in mathematical programming techniques for optimizing process systems under uncertainty.[128,129] Applequist et al. (2000) as well mentioned the importance of uncertainty and risk in designing and managing supply chains in chemical manufacturing.[130] Turning to research specifically on pharmaceutical manufacturing, Chhatre et al. (2008) represented uncertainty and utilized GSA in purification processes for polyclonal antibodies.[131] Lakhdar and Papageorgiou (2008) presented a specific mathematical programming approach for planning biopharmaceutical manufacturing under uncertainty.[132] Lakerveld et al. (2013) applied sensitivity analysis to identify parameters affecting critical quality attributes in a model-based design of a plant-wide control strategy for a continuous pharmaceutical plant.[133] Most recently, Li and Venkatasubramanian (2016) proposed a Bayesian approach for predicting the performance of upcoming batches by leveraging historical data in biopharmaceutical DS manufacturing.[134]

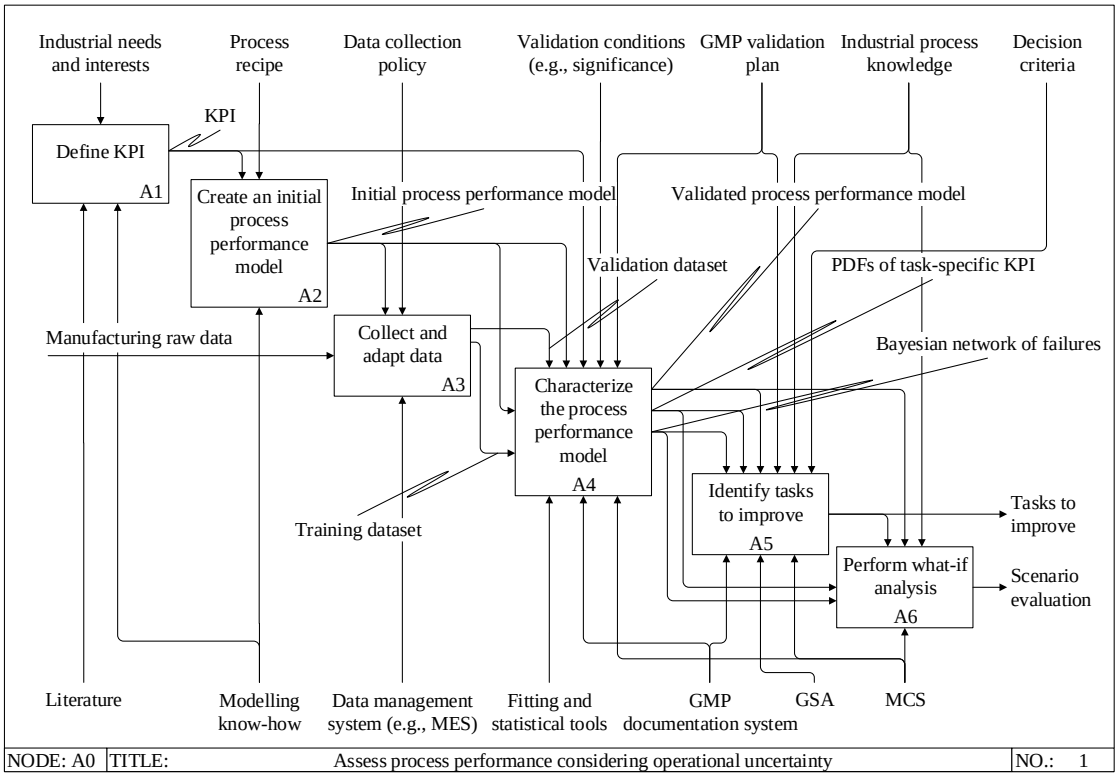
Beyond the development and application of the individual methods reviewed above, some authors have advanced the research to create integrated methodologies or frameworks for process performance improvement. Garcia-Munoz and Mercado presented a method of raw materials selection using mixed integer nonlinear programming and multivariate latent variable regression models in small-molecule DP manufacturing.[135] Casola et al. (2015) also focused on small-molecule DS manufacturing, and proposed a systematic procedure of process modeling for process retrofitting.[136] Eberle et al. (2014) presented an approach based on Monte Carlo Simulation (MCS) to reduce the lead time in DP manufacturing activities, such as compounding, filling, inspection, and quality assurance.[104] However, these contributions share a shortcoming, in that uncertainty of the process and the constraints of GMP are not in the foreground of the methodology, even though these factors are inevitable challenges in process performance improvement.

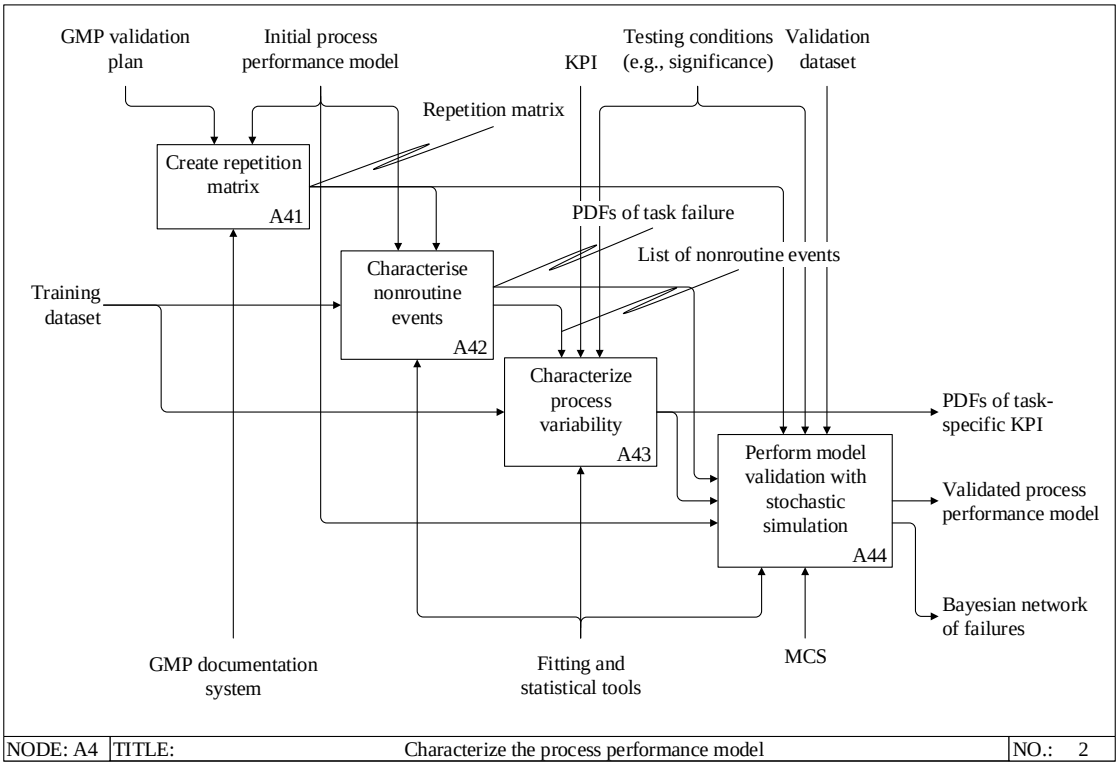
In this chapter, a methodology for the assessment of process performance in biopharmaceutical DP manufacturing that integrates operational uncertainty and GMP regulations is presented. The methodology is described as an activity model using the type 0 Integrated Definition (IDEF0) functional modeling method, which systematically interconnects information, tools, and activities.[137] In executing the methodology, a hybrid stochastic-deterministic model for describing operational uncertainty and GMP-related factors that can affect process performance is created. MCS is one of the key mechanisms used to propagate and reflect operational uncertainty in the assessment result. The combination of stochastic simulation and GSA—e.g., Partial Rank Correlation Coefficients (PRCCs)—quantitatively measures the effects of individual contributions to the overall process performance. Tailored indicators were developed to consider factors that are essential in decision-making on the shop-floor level, such as the effort required for process validation, or the risk of overestimating the benefits. As a case study, the methodology was applied to a CIP/SIP process in a commercial facility for biopharmaceutical DP manufacturing. The author have presented a previous version of the methodology in part in a six-page conference proceedings paper,[138] but this paper presents the complete work with full details.

4.2 Methodology

Figure 4.1 displays the developed methodology using IDEF0, a general explanation of which is presented in the Appendix (see Appendix D.2 IDEF0). The top activity (top), A0, takes raw manufacturing data as an input (the arrow from the left to the box) and delivers "tasks to improve" and "scenario evaluation" as outputs (arrows from the box to the right). The activity defines literature, modelling know-how, the data management system, fitting and statistical tools, the GMP documentation system, GSA, and MCS as mechanisms (the arrows from the bottom to the box), and industrial needs and interests, process recipes, the data collection policy, validation conditions, the GMP validation plan, industrial process knowledge, and decision criteria as controls (arrows from the top to the box). Activity A0 (upper middle, **Figure 4.1**) consists of six subactivities, namely "define Key Performance Indicators (KPIs) (A1)," "create an initial process performance model (A2)," "collect and adapt data (A3)," "characterize the process performance model (A4)," "identify tasks to improve (A5)," and "perform what-if analysis (A6)."




Industrial needs and interests
Process recipe
Data collection policy
Validation conditions (e.g., significance)
GMP validation plan
Industrial process knowledge
Decision criteria
KPI
Define KPI
A1
Create an initial process performance model
A2
Initial process performance model
Validated process performance model
Validation dataset
PDFs of task-specific KPI
Manufacturing raw data
Collect and adapt data
A3
Bayesian network of failures
Characterize the process performance model
A4
Identify tasks to improve
A5
Tasks to improve
Training dataset
Perform what-if analysis
A6
Scenario evaluation
Literature
Modelling know-how
Data management system (e.g., MES)
Fitting and statistical tools
GMP documentation system
GSA
MCS
NODE: A0
TITLE:
Assess process performance considering operational uncertainty
NO.: 1


GMP validation plan
Initial process performance model
KPI
Testing conditions (e.g., significance)
Validation dataset
Repetition matrix
Create repetition matrix
A41
PDFs of task failure
List of nonroutine events
Training dataset
Characterise nonroutine events
A42
Characterize process variability
A43
PDFs of task-specific KPI
Perform model validation with stochastic simulation
A44
Validated process performance model
Bayesian network of failures
GMP documentation system
Fitting and statistical tools
MCS
NODE: A4
TITLE:
Characterize the process performance model
NO.: 2

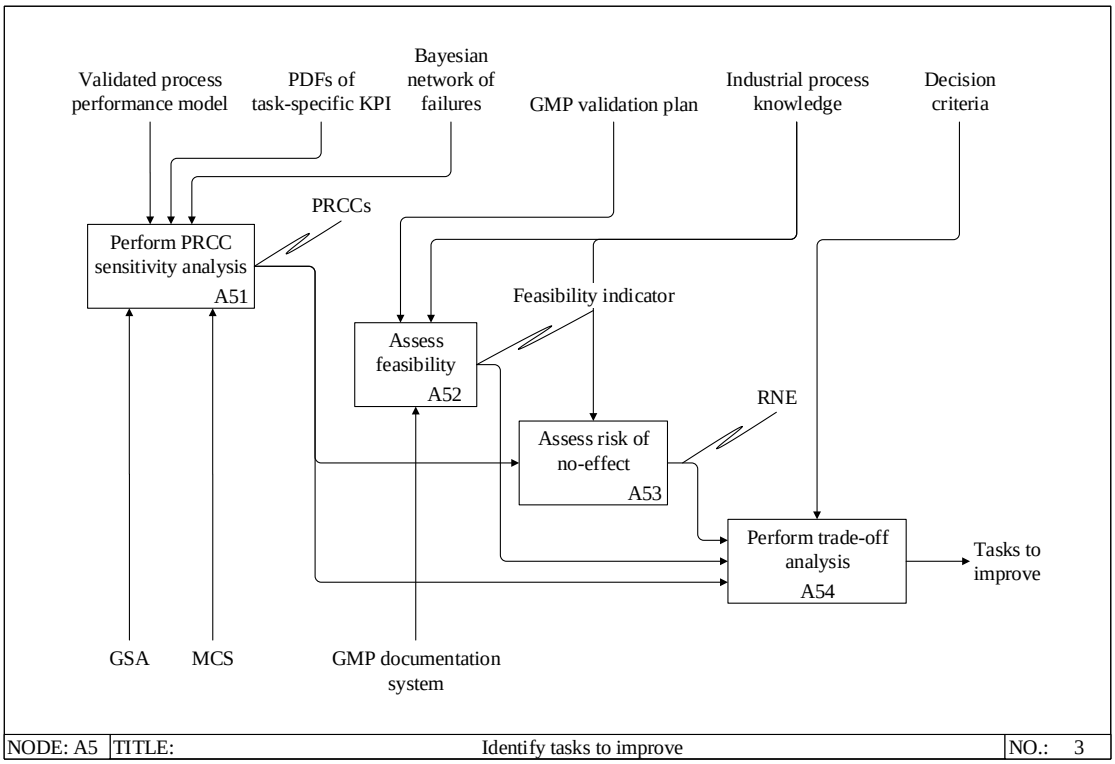


Figure 4.1 IDEF0 representation of the process performance assessment methodology. The layers are shown hierarchically: The top layers (top), the A0 (upper middle), the A4 (lower middle), the A5 (bottom) layer

4.2.1 Define KPI (activity A1)

In activity A1, KPI is defined according to industrial needs and interests. Examples of KPIs in biopharmaceutical DP could be runtime, product loss, and energy consumption, which reflect productivity, capacity, and environmental impact, respectively. Literature and modeling know-how can be referred to as mechanisms to define an appropriate KPI.

4.2.2 Create an initial process performance model (activity A2)

In activity A2, the initial process performance model is created through modeling know-how under the control of the process recipe as well as the previously defined KPIs. Usually, the nature of process models is either deterministic—e.g., first-principles models—or stochastic—e.g., black box models or metamodels—depending on the KPIs. However, to account for operational uncertainty, this work introduces the use of a so-called "hybrid stochastic-deterministic model"[139] that can be formulated as:

$$KPI = \sum_{k=1}^{K} (1 + q_k) \cdot KPI_k \tag{4.1}$$

where K represents the total number of tasks, q_k is the counter representing the number of times task k is repeated because of failures, and KPI_k is the KPI specific to task k. Depending on the level of automation of the task, KPI_k can be either distributed or constant. The use of Eq. (4.1) is suitable for KPIs that describe additive quantities, such as time, mass, or energy.

4.2.3 Collect and adapt data (activity A3)

In activity A3, raw manufacturing data are collected from a data management system such as a Manufacturing Execution System (MES), sorted, and adapted to the purpose of the model. The activity removes extreme outliers and noise from the raw dataset under the control of the data collection policy and delivers two labeled datasets, namely the validation and training datasets. The data collection policy also controls the amount of data or the number of batches that are required to conduct the analyses.

4.2.4 Characterize the process performance model (activity A4)

In activity A4, the initial model is characterized and validated. The activity defines as inputs the initial process performance model from activity A2 and the training dataset from A3, and the outputs are the validated process performance model, the Probability Distribution Functions (PDFs) of task-specific KPI, and the Bayesian network of failures. The activity uses the GMP documentation system, fitting and statistical tools, and MCS as mechanisms, and is controlled by the validation dataset, the initial model, data collection policy, GMP validation plan, KPI, and validation conditions. Generally, the GMP validation plan guides the process validation, but it also provides the validated repetition sequence in the case of failure.

4.2.4.1 Create repetition matrix (activity A41)

In activity A41, the validated repetition sequence of task k is described by the repetition matrix $[K \times K]$, the element of which is $\boldsymbol{R}_{k',k} = 0$ or 1. The matrix is created according to the GMP validation plan extracted from

the GMP documentation system and the initial process performance model. A value of 1 indicates that task k' must be repeated in the case of the failure of task k, and a value of 0 means there is no need to repeat task k'.

4.2.4.2 Characterize nonroutine events (activity A42)

In activity A42, nonroutine events are detected and summarized as a list under the control of the initial process performance model and repetition matrix. After statistically testing that the data in the training dataset are independent and identically distributed for each task, the Bernoulli PDFs are fitted to the training dataset. The fitted PDF, $B(f_k, p_k)$ (hereafter termed the PDF of task failure), which represents the occurrence of failures for each task k, is shown by:

$$B(f_k, p_k) = \begin{cases} 1 - p_k & \text{for } f_k = 0 \\ p_k & \text{for } f_k = 1 \end{cases} \tag{4.2}$$

The Boolean variable $f_k \in \{0,1\}$ represents the success or the failure of task k by 0 or 1, respectively. The parameter p_k represents the prior probability of the failure of task k, and is calculated by:

$$p_k = \frac{\text{Number of nonroutine events in task } k}{\text{Total number of events in task } k} \tag{4.3}$$

4.2.4.3 Characterize the process variability (activity A43)

In activity A43, the PDFs of task-specific KPI, $\hat{f}(KPI_k)$, are calculated. Continuous PDFs are fitted to the training dataset using fitting and statistical tools to describe the process variability for a task. The choice of the PDF can be either parametric—e.g., normal, lognormal, or Weibull—or nonparametric—e.g., kernel[140]—depending on the characteristics of the KPI. After fitting, goodness-of-fit is statistically tested by a Kolmogorov–Smirnov test with a 5% significance level.[141] The PDFs of the task-specific KPI are used in activity A44 as the sampling pools for estimating the overall process KPIs by means of the initial process performance model (see Eq. (4.1)).

4.2.4.4 Perform model validation with stochastic simulation (activity A44)

MCS is used in activity A44 to validate the initial process performance model created in activity A2. In this work, the concept of a "failure layer" is introduced to represent consecutive failures in a batch and to count the position of such failures. The index j specifies the failure layer of task k, and the variables $p_{k,j}$ and $f_{k,j}$ indicate

the probability of failure and the Boolean success/failure variable, respectively. Because failures are consecutive, the calculation of the probability of failure $p_{k,j}$ at failure layer j is dependent on $(j-1)$. The probability $p_{k,j}$ is calculated from the original probability distributions $B(f_k, p_k)$ as shown by the progression in Eq. (4.4).

$$\begin{cases} p_{k,j} = p_{k,1} \cdot u^{(k)}_{K-k,j-1} & \text{for } j \geq 2 \\ p_{k,1} = p_k & \text{for } j = 1 \end{cases} \tag{4.4}$$

The term $u^{(k)}_{K-k,j-1}$ describes the probability of encountering a failure in the tasks from k to K that requires the repetition of task k in failure layer $(j-1)$; and $u^{(k)}_{K-k,j-1}$ is the last term of the progression $u^{(k)}_{i,j-1}$ in Eq. (4.5), where i is the counter in the equation. The probability of the disjunction of multiple events, explained by Eq. (4.6) where A and B represent two general events, is calculated by Eq. (4.5).

$$\begin{cases} u^{(k)}_{i,j-1} = \boldsymbol{R}_{k,k+i} \cdot p_{k+i,j-1} + u^{(k)}_{i-1,j-1} - \boldsymbol{R}_{k,k+i} \cdot p_{k+i,j-1} \cdot u^{(k)}_{i-1,j-1} \\ u^{(k)}_{0,j-1} = p_{k,j-1} \qquad \text{for } i = 1,2,3,\dots,K-k\,;\, j \geq 2 \end{cases} \tag{4.5}$$

$$P(A \vee B) = P(A) + P(B) - P(A)P(B) \tag{4.6}$$

The matrix element $\boldsymbol{R}_{k,k+i}$ in Eq. (4.5) integrates GMP-related constraints into the calculation. The probability distribution is defined as $M(j \cdot f_{k,j},\ p_{k,j})$, which represents the sample pool for the failure counter q_k (see Eq. (4.1)). It is the multinomial probability distribution of consecutive failures weighted by the failure layer j. In practice, the probability distribution M is used to create the Bayesian network of failures that represents the structure of failure sequences and the probabilities of failure occurrence inside the structure.

The stochastic simulation is performed with Eq. (4.1), where KPI_k and q_k are sampled—e.g., 10,000 iterations—from the PDFs created above, $\hat{f}(KPI_k)$ and $M(j \cdot f_{k,j},\ p_{k,j})$, respectively. The initial model is validated by the Kolmogorov–Smirnov test for the simulation outcome using the validation dataset, at a 5% significance level. If the validation is unsuccessful, new information is required regarding the data and the model, and this activity is repeated until validation is achieved. The validated model can then be used to estimate the overall process KPI from the task-specific KPI incorporating operational uncertainty—i.e., nonroutine events and process variability.

4.2.5 Identify improvement targets (activity A5)

Activity A5 is the key activity that identifies tasks to improve. The mechanisms are the GMP documentation system, fitting and statistical tools, GSA, and MCS; the controls are the validated process performance model, PDFs of task-specific KPI, Bayesian network of failures, GMP validation plan, industrial knowledge, and decision criteria. This activity considers both the effect of each task on the overall process performance and the essential factors for decision making on the shop-floor level.

4.2.5.1 Perform PRCC sensitivity analysis (activity A51)

Activity A51 conducts stochastic GSA using PRCCs, which delivers a ranking of the tasks *k* according to the impact of each KPI_k on the overall process KPIs. Here, the parameters that are subject to sensitivity analysis, namely KPI_k, are defined as changeable parameters.

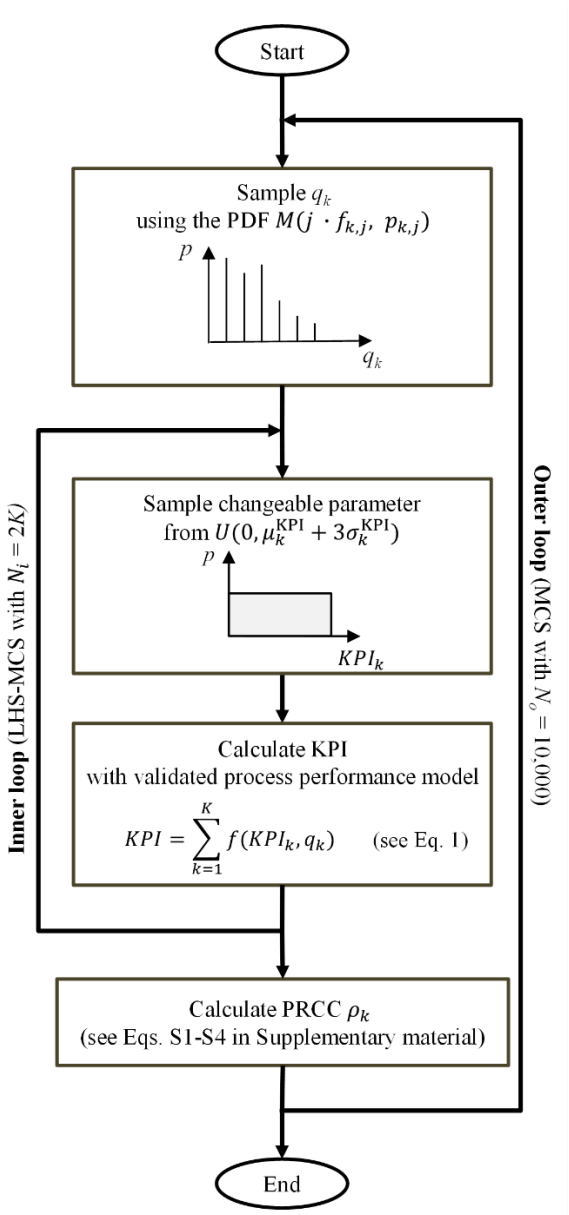


Figure 4.2 Approach for calculating PRCCs ρ_k using two nested loops.

The coefficient ρ_k for task k is calculated using two nested loops, as shown in **Figure 4.2**. In this approach, variables representing operational uncertainty and changeable parameters are independently sampled in the outer and inner loops, respectively. The outer loop is iterated N_o times, and in this loop the values of q_k are sampled using M in the Bayesian network of failures. These values are used as inputs to the inner loop, where the sensitivity analysis is performed by executing Latin Hypercube Sampling–MCS (LHS–MCS) with the validated process performance model (iterated $N_i = 2K$ times with the counter n_i; see section B.2 PRCC

calculation in the appendix).[142,143] The values of the changeable parameters are sampled from uniform probability distributions $U(0, \mu_k^{\text{KPI}} + 3\sigma_k^{\text{KPI}})$, where μ_k^{KPI} and σ_k^{KPI} represent the mean and standard deviation of $\hat{f}(KPI_k)$, respectively. The PRCC ρ_k is the result of the stochastic GSA and is defined as the output of this activity. This approach differentiates the impact on process performance of design—e.g., the nominal run time assigned to a specific task in the design—from the impact caused by operational uncertainty—e.g., repetition of tasks because of nonroutine events.

4.2.5.2 Assess feasibility (activity A52)

In activity A52, the feasibility indicator Φ_k is defined, and is used to assess the feasibility of modifying task k, considering the effort to (re-)validate the modified process. Both the GMP validation plan and industrial process knowledge provide a basis for estimating this effort, with the primary consideration being product quality. Lower feasibility values are assigned to tasks that play significant roles in determining product quality and thus are highly controlled by GMP regulations, and higher values are assigned to tasks that can be modified more freely. The indicator can cover other factors, such as the time or cost of implementing change, and company strategies and priorities. The scaling of the indicator is case dependent, and is established in a relative manner by investigating which changes are more feasible relative to others.

4.2.5.3 Assess the risk of no-effect (activity A53)

Activity A53 assesses the risk of overestimating the benefits of modifying the task, here termed the "Risk of No-Effect" (RNE). RNE is assessed by evaluating ϵ, which represents the normalized PRCC, as shown in Eq. (4.7).

$$\epsilon = \frac{\rho_k}{Mo(\rho_k)} \tag{4.7}$$

By using Eq. (4.7), it is possible to calculate the probability that activity A51 predicts a significantly lower ρ_k than the most frequent ρ_k, which is represented by the mode $Mo(\rho_k)$. The normalized PRCC ϵ is associated with its occurrence probability for every changeable parameter, then tuples comprising these ϵ values and their associated probabilities are used to characterize the RNE. For instance, the low-ϵ, high-probability tuple suggests that modifying the task has a high RNE, which is not preferable. In contrast, the high-ϵ, high-

probability tuple indicates that the modification of the task would very likely result in significant process performance improvement. Industrial process knowledge, which is a control for this activity, is utilized to define the classifications of high versus low ϵ values and probabilities.

4.2.5.4 Perform trade-off analysis (activity A54)

In activity A54, tasks to improve are recognized based on the outputs of activities A51, A52, and A53. Trade-off analysis is performed to identify the tasks that have high values of both ρ_k and Φ_k. The identified tasks are further screened according to the values of ϵ. The activity is controlled by the decision criteria that define the limiting acceptance values for feasibility and RNE indicators, and identifies the tasks to improve as the output.

4.2.6 Perform what-if analysis (activity A6)

In activity A6, what-if analysis is performed to evaluate alternative scenarios, for example, technical changes in the facility or process redesign. The what-if analysis is conducted with the validated process performance model using MCS, in which the sampling pool $\hat{f}(KPI_k)$ and/or the Bayesian network of failures are modified. Either the outcome of A54 or industrial process knowledge can initiate the analysis, as controls for activity A6. This activity produces scenario evaluations as the output. These play a particularly important role in the creation of a well-assessed business case before investing in implementing any process changes.

4.3 Case study

The methodology was applied to the facility of F. Hoffmann–La Roche Ltd. More specifically, a process performance assessment on two, *intra*-CIP/SIP and *post*-CIP/SIP, was conducted, respectively. The *intra*-CIP/SIP consisted of 186 tasks that were subdivided into ten main blocks. The *post*-CIP/SIP consisted of 180 tasks subdivided into nine blocks. **Table 4.1** lists every process block that contained a series of automated and manual tasks, such as “rinsing the piping for 10 min with DW” (*B6*) or “changing the piping format” (*D2*). The *intra*-CIP/SIP and *post*-CIP/SIP differed in the presence or absence of the format-change process block (*D*) and a part of the impermeability testing (*E9*), respectively.

Table 4.1 Table summarizing the tasks in blocks *A* to *J*. Block *D* and task *E9* are only performed in *intra*-CIP/SIP.

ID	Block description	Tasks
A	Preparation of the CIP/SIP	*A1–A7*
B	Rinsing the piping with DW	*B1–B30*
C	Filter integrity test	*C1–C34*
D	Format change (only in *intra*-CIP/SIP)	*D1–D5*
E	Impermeability testing of the filling needles	*E1–E8* (*E9*)
F	Rinsing the piping with WFI	*F1–F27*
G	Sterilization of the system	*G1–G18*
H	Drying and cooling of the piping	*H1–H21*
I	Integrity testing of the production filter after SIP	*I1–I32*
J	End of CIP/SIP	*J1–J3*

4.3.1 Define KPI (activity A1)

In DP manufacturing, it is necessary to maintain sufficient production capacity to supply life-saving products to patients in an agile way without delay. Therefore, activity A1 defined the total run time of the CIP/SIP process as the main KPI for the overall process.

4.3.2 Create an initial process performance model (activity A2)

The overall process KPI (total run time) was modeled according to the process recipe, which defined the structure and the sequence of the tasks in the process, and is shown in Eq. (4.8):

$$T^{\mathrm{CIP/SIP}} = \sum_{k=1}^{K} \left[t_k^{\mathrm{task}} + t_k^{\mathrm{corr}} \cdot c_k^{\mathrm{rep}} + t_k^{\mathrm{rem}} \cdot c_k^{\mathrm{rep}} + \sum_{k' \in Rep(k)} \left(t_{k'}^{\mathrm{task}} \cdot c_{k'}^{\mathrm{rep}} \right) \right]; \qquad c_k^{\mathrm{rep}} = f(p_k, \dots, p_K) \tag{4.8}$$

The variable $T^{\mathrm{CIP/SIP}}$ represents the total run time of the CIP/SIP process, and the variable t_k^{task} is the run time of task k. If task k fails, time is wasted for as long as that task remains corrupted, represented as t_k^{corr}, and further time is needed to remedy the failure, represented as t_k^{rem}. The integer c_k^{rep} accounts for the number of failures of task k, and it is a function of the failure probability p_k. The internal summation term represents the time invested in repeating tasks because of the failure of task k. The set $Rep(k)$ contains the indices k' of the multiple tasks that must be repeated in the case of failure in task k. Lastly, the variable K is the total number of tasks that must be performed correctly in *intra*-CIP/SIP ($K = 186$) and *post*-CIP/SIP ($K = 180$).

4.3.3 Collect and adapt data (activity A3)

The raw manufacturing data related to the runtime of the tasks were collected from MES (the data management system) for 137 commercial batches that were included in 50 campaigns. The raw manufacturing data were assumed to be independent of *intra*- or *post*-CIP/SIP and batch, and the data were sorted by tasks according to the index k. To test the assumption of independence, the correlation coefficients between task run times were calculated, which confirmed the uncorrelated nature of the data.

The data that were sorted by tasks were adapted to the initial model; by Eq. (4.8), run times were calculated as the difference between the time flags recorded in MES as shown in **Table 4.2**.

Table 4.2 Example of raw data adaptation.

Raw data from the MES		**Adapted data used for creating the validation and training datasets**	
Date	Task ID	New Task ID	Duration (h)
Nov-06-15 15:44:00	"2100"	A1	1.02
Nov-06-15 16:45:00	"2110"	A2	0.17
Nov-06-15 16:50:05	"3000"	B1	0.65
Nov-06-15 17:29:05	"3100"	B2	…

For simplicity, a continuous production timeline—i.e., 24/7 operations—was assumed. Whenever obvious, outliers were removed. The adapted data were split between the training and the validation datasets, which contained 90% and 10% of the original dataset, respectively.

4.3.4 Characterize the process performance model (activity A4)

4.3.4.1 Create repetition matrix (activity A41)

The repetition matrix $\boldsymbol{R}^{\mathrm{CIP/SIP}}$, was created to describe the patterns of validated repetition of the tasks. Eqs. (4.9) and (4.10) summarize the matrices $\boldsymbol{R}^{\mathrm{CIP/SIP}}$, non-zero elements of which ($\boldsymbol{R}_{X,Y}^{\mathrm{CIP/SIP}}$) show the block X that must be repeated in the case of failure in block Y for *intra*-CIP/SIP and *post*-CIP/SIP, respectively.

$$\boldsymbol{R}^{intra-\mathrm{CIP/SIP}} = \begin{bmatrix} \boldsymbol{R}^{A} & 0 & 0 & & & & & & & \\ 0 & \boldsymbol{R}^{B} & 0 & & & 0 & & & 0 & \\ 0 & 0 & \boldsymbol{R}^{C} & & & & & & & \\ & & & \boldsymbol{R}^{D} & 0 & 0 & 0 & 0 & \boldsymbol{R}^{E} & 0 \\ & 0 & & 0 & \boldsymbol{R}^{E} & 0 & & 0 & \boldsymbol{R}^{F} & 0 \\ & & & 0 & 0 & \boldsymbol{R}^{F} & 0 & \boldsymbol{R}^{G} & \boldsymbol{R}^{G} & 0 \\ & & & 0 & 0 & 0 & \boldsymbol{R}^{G} & \boldsymbol{R}^{H} & \boldsymbol{R}^{H} & 0 \\ & 0 & & & 0 & & & 0 & \boldsymbol{R}^{I} & 0 \\ & & & & & & & 0 & 0 & \boldsymbol{R}^{J} \end{bmatrix} \tag{4.9}$$

$$\boldsymbol{R}^{post-\mathrm{CIP/SIP}} = \begin{bmatrix} \boldsymbol{R}^A & 0 & 0 & & & & & & \\ 0 & \boldsymbol{R}^B & 0 & & 0 & & & 0 & \\ 0 & 0 & \boldsymbol{R}^C & & & & & & \\ & & & \boldsymbol{R}^E & 0 & 0 & 0 & \boldsymbol{R}^E & 0 \\ & 0 & & 0 & \boldsymbol{R}^F & 0 & 0 & \boldsymbol{R}^F & 0 \\ & & & 0 & 0 & \boldsymbol{R}^G & \boldsymbol{R}^G & \boldsymbol{R}^G & 0 \\ & & & & & & \boldsymbol{R}^H & \boldsymbol{R}^H & 0 \\ & 0 & & & 0 & & 0 & \boldsymbol{R}^I & 0 \\ & & & & & & 0 & 0 & \boldsymbol{R}^J \end{bmatrix} \tag{4.10}$$

Usually, the block where the failure occurred is repeated; however, in the case of failures in quality-determining blocks such as filter integrity tests (block *I*), the GMP validation plan—i.e., the general GMP requirement—requires the repetition of multiple previous blocks. The general matrix element $\boldsymbol{R}^x$ in Eqs. (4.9) and (4.10) is an upper-triangular unit matrix of dimensions [# tasks in block x × # tasks in block x] as shown in Eq. (4.11). A matrix element $\boldsymbol{R}^x_{k',k}$ equal to 1 indicates that task k' must be repeated in the case of failure in task k in block x.

$$\boldsymbol{R}^x = \begin{bmatrix} 1 & 1 & \dots & 1 & 1 \\ 0 & 1 & \dots & 1 & 1 \\ 0 & 0 & \dots & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.11}$$

$\boldsymbol{R}^{\mathrm{CIP/SIP}}$ was used to create the set $Rep(k)$ that contains the indices k' of matrix elements equal to 1 for task k (see Eq. (4.8)).

4.3.4.2 Characterize nonroutine events (activity A42)

According to Eq. (4.8), this activity characterizes routine and nonroutine events, here represented by $op \in \{\mathrm{task}, \mathrm{rem}, \mathrm{corr}\}$. The PDFs of task failure $B(f_k, p_k)$ (see Eq. (4.3)) were fitted to the training dataset, by estimating p_k as shown by Eq. (4.2).

4.3.4.3 Characterize the process variability (activity A43)

The run times t_k^{task}, t_k^{rem}, and t_k^{corr} were estimated by fitting the kernel PDFs to the training dataset, to produce the PDFs of task-specific KPI, $\hat{f}\left(t_k^{op}\right)$. Kernel functions were used because of their versatility. The goodness-of-fit was statistically demonstrated with the Kolmogorov–Smirnov test at a 5% significance level. KPIs involving time are only defined for the interval $[0, +\infty]$; hence, to ensure the physical feasibility of the PDFs, the interval shown in Eq. (4.12) was enforced in the estimation calculation.

$$\begin{cases} \hat{f}\left(t_k^{op}\right) = 0 & \text{for } t_k^{op} < 0 \\ \hat{f}\left(t_k^{op}\right) \geq 0 & \text{for } t_k^{op} \geq 0 \end{cases} \tag{4.12}$$

4.3.4.4 Perform model validation with stochastic simulation (activity A44)

The initial process performance model was validated through MCS. To calculate the number of failures represented by the parameter c_k^{rep}(see Eq. (4.8)), the algorithm was applied that is shown in the section B.1 Algorithm for calculating c_k of the appendix was applied.

Contrary to the work presented by Cadini and Gioletta,[127] the number of failures c_k^{rep} was not directly generated from the PDF $M(j \cdot f_{k,j},\ p_{k,j})$, but was stochastically calculated with a failure-after-failure approach. To prevent the intensive explicit formulation of the joint probabilities $p_{k,j}$ (see Eq. (4.4)), the infinitely large Bayesian network of failures represented by the PDF $M(j \cdot f_{k,j},\ p_{k,j})$ was approximated with MCS. The approximated network was the output of this activity. This approach reduced the computational effort needed to model large systems—i.e., $K = 180$ or 186—and infinite event scales—i.e., $j \to \infty$, as was the case in this study. **Figure** 4.3 shows three examples of binary-element failure matrix $\boldsymbol{F}$ generated with the presented algorithm.

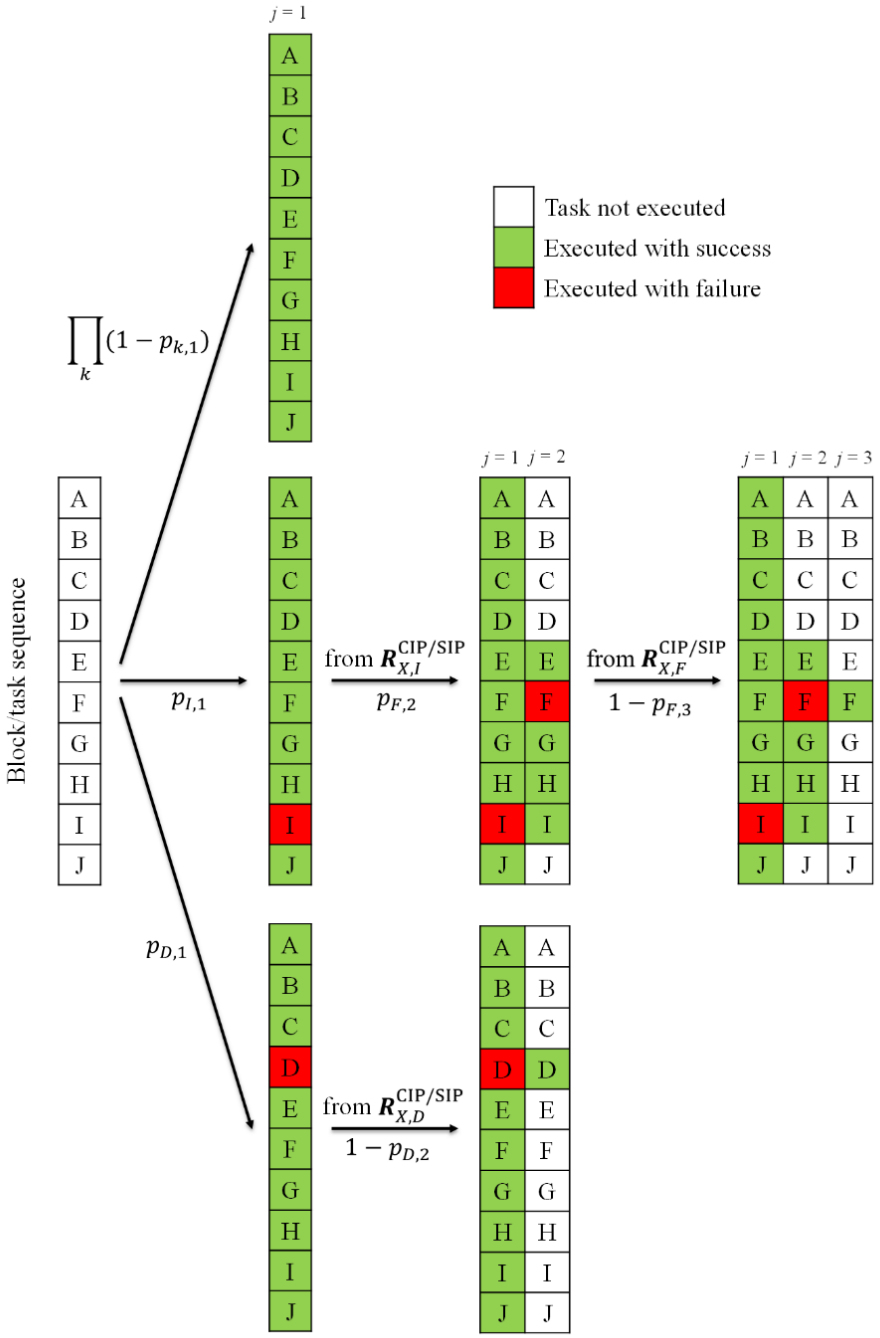


Figure 4.3 Example of the failure matrix F constructed using the concept of failure layer.

After the Bayesian network of failures and c_k^{rep} were determined, the variables t_k^{task}, t_k^{rem}, and t_k^{corr} were independently sampled from the PDFs of the task-specific KPI, $\hat{f}(t_k^{op})$. The outcome of the activity was the model, which was validated with the Kolmogorov-Smirnov test, yielding p-values of 0.15 and 0.05 for *intra*- and *post*-CIP/SIP, respectively.

4.3.5 Identify improvement targets (activity A5)

4.3.5.1 Perform PRCC sensitivity analysis (activity A51)

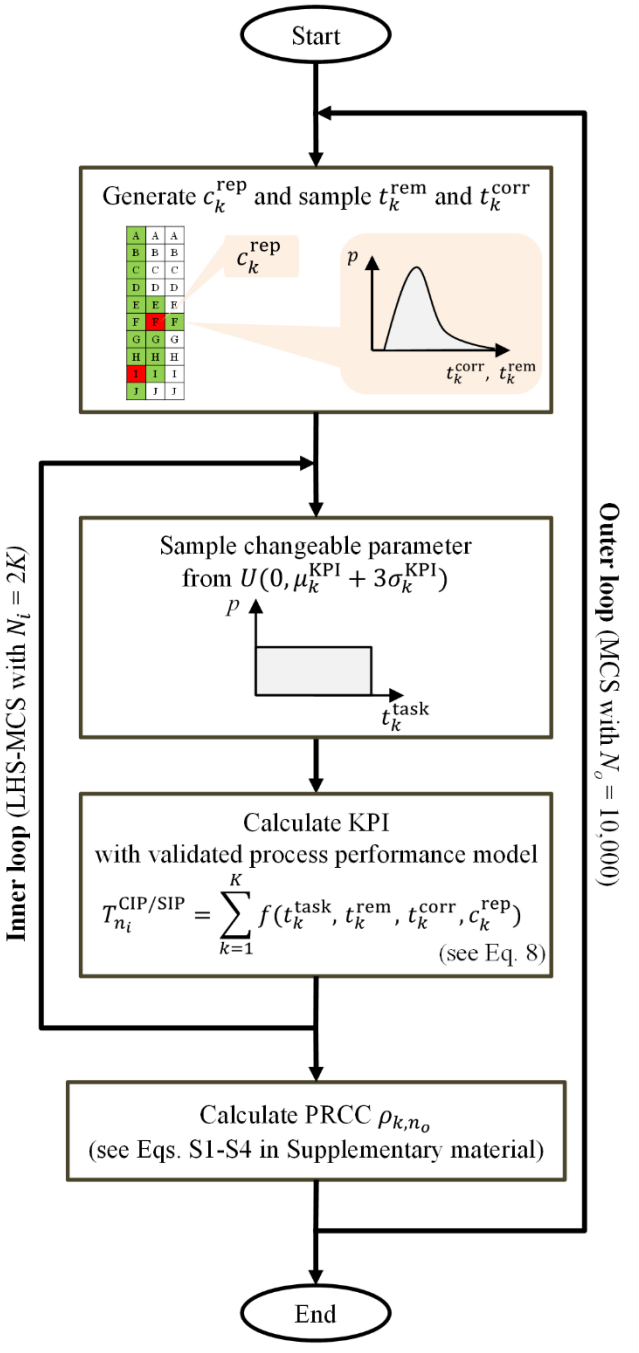


Figure 4.4 Approach for calculating PRCCs ρ_k using two nested loops, as applied in the case study.

The approach using two nested loops to conduct the stochastic PRCC sensitivity analysis is shown in **Figure 4.4**. To analyze the contribution of the design of each task to the total process KPI (here, total runtime), t_k^{task}

(K = 180 and 186) were defined as changeable parameters. The variables t_k^{corr}, t_k^{rem}, and c_k^{rep} accounted for the operational uncertainty, and were generated using the PDFs of the task-specific KPI, $\hat{f}$, from activity A43 and the approximated Bayesian network of failures from activity A44. At every iteration n_o of the outer loop ($N_o = 10{,}000$ iterations), one operational layout—i.e., a set of values for t_k^{corr}, t_k^{rem}, and c_k^{rep}—was generated and was used in the inner loop to calculate the total KPI. Here, LHS–MCS ($N_i = 2K$ iterations) sampled changeable parameters from uniform probability distributions $U\left(0,\ \mu_k^{\mathrm{task}} + 3 \cdot \sigma_k^{\mathrm{task}}\right)$ using the performance model (see Eq. (4.8)) to calculate $T_{n_i}^{\mathrm{CIP/SIP}}$. The coefficient ρ_{k,n_o} was calculated each time the inner loop was completed. See section B.2 PRCC calculation in the appendix for details.

The result of the sensitivity analysis is shown in **Figure 4.5**, where the PRCCs ρ_k of the 20 most important tasks are highlighted for *post*-CIP/SIP and *intra*-CIP/SIP on the left and right, respectively.

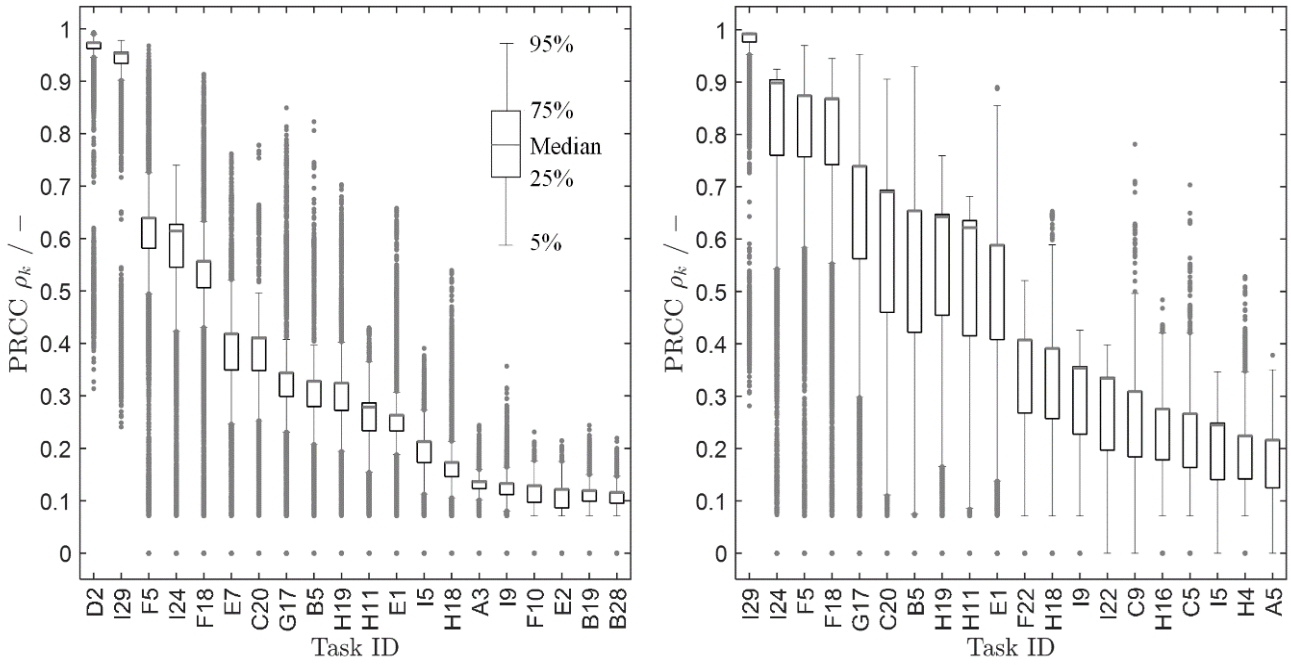


Figure 4.5 Result of PRCC analysis for the 20 most important tasks in *intra*-CIP/SIP (left) and *post*-CIP/SIP (right).

It can be observed, by comparing *intra*-CIP/SIP and *post*-CIP/SIP in **Figure 4.5**, that the former shows narrower probability distributions of ρ_k than the latter. For both *intra*-CIP/SIP and *post*-CIP/SIP, the probability of failure p_k was generated from the same PDF *B* for the tasks in common. One important difference between *intra*-CIP/SIP and *post*-CIP/SIP was the presence of block *D* in *intra*-CIP/SIP, which was particularly time intensive.

PRCC presents the relative impact of tasks by comparing the impact of one task with that of the others within the process. For these reasons, the time-intensive tasks in block *D* such as *D2* significantly lowered the mean $\bar{\rho}_k$ and narrowed the probability distributions of the PRCCs in *intra*-CIP/SIP.

4.3.5.2 Assess feasibility (activity A52)

The feasibility of changing tasks was assessed by creating a feasibility indicator Φ_k that was developed through brainstorming sessions with subject matter experts on the process. The experts were asked to quantify the feasibility of process modification in a range from 0 to 1, considering quality risk, revalidation efforts, and corporate strategies and priorities. An Φ_k equal to 0 indicated that the task was completely unmodifiable, and 1 indicated that the task could be freely modified.

4.3.5.3 Assess risk of no-effect (activity A53)

The risk of no-effect was assessed by calculating the normalized PRCC ϵ (see Eq. (4.7)) and the results are presented in **Figure 4.6**.

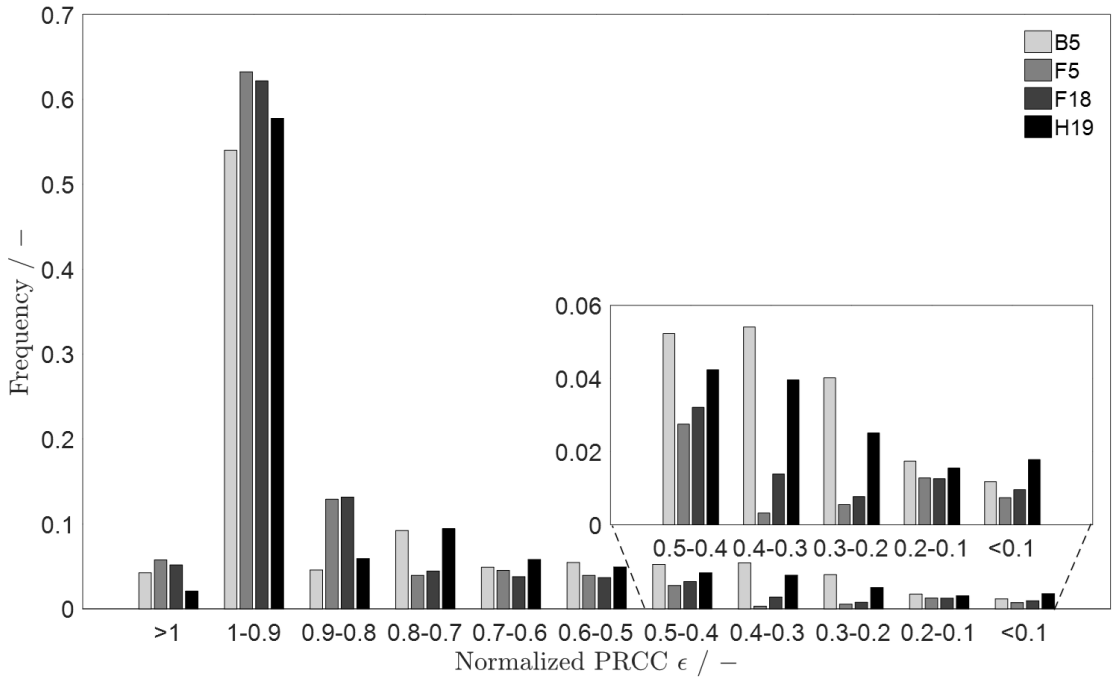


Figure 4.6 RNE assessment results for the tasks *B5*, *F5*, *F18*, and *H19* belonging to *post*-CIP/SIP in the case study.

The probability distribution of the normalized PRCC is presented in **Figure 4.6**, where ϵ is discretized into intervals of [0,1] and the corresponding probability is shown on the *y*-axis. For explanatory purposes, **Figure**

4.6 only displays RNEs for the tasks *B5*, *F5*, *F18*, and *H19* in *post*-CIP/SIP; the results of the remaining tasks are shown in **Figure B.1** in the appendix. The probability distribution shows tailing in areas of low ϵ values, which suggests a possible overestimation of the PRCC in the presented tasks. In this case, values of ϵ lower than 0.8 were defined as "low-ϵ". If a task shows low ϵ and high probability values, then the effect of improving this specific task would be overestimated—i.e., high RNE.

4.3.5.4 Perform trade-off analysis (activity A54)

"Tasks to improve" were identified; the trade-off analysis of Φ_k and $\bar{\rho}_k$ is shown in **Figure 4.7**. In this case study, $\Phi_k \geq 0.5$ and $\bar{\rho}_k \geq 0.5$ were applied as decision criteria for specifying the tasks in the "high-impact, high-feasibility" zone, which were the ones of most interest. In **Figure 4.7**, tasks *D2*, *F5*, and *F18* and *B5*, *F5*, *F18*, and *H19* lay in the "high–impact, high–feasibility" zone, for *intra*-CIP/SIP (left) and *post*-CIP/SIP (right), respectively. For these tasks, RNE was assessed. From the magnified inset in **Figure 4.7**, task *B5* was classified as a task with high RNE, and therefore the modification of this task was considered less preferable. Tasks to improve—i.e., a general output of the methodology—were the tasks that passed the screenings, namely *D2*, *F5*, and *F18* for *intra*-CIP/SIP and *F5*, *F18*, and *H19* for *post*-CIP/SIP.

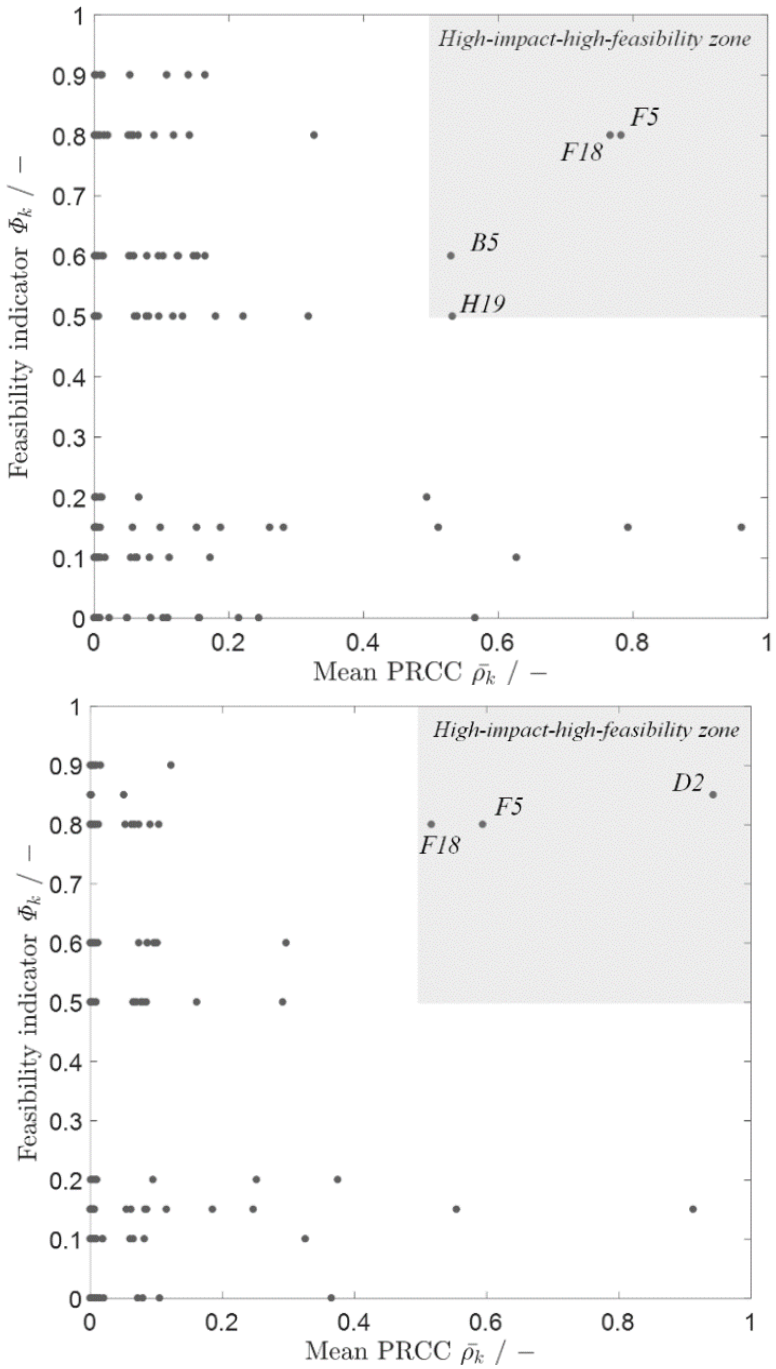


Figure 4.7 Trade-off between the PRCC and the feasibility indicator for *post*-CIP/SIP (top) and *intra*-CIP/SIP (bottom).

4.3.6 Perform what-if analysis (activity A6)

In this activity, three different improvement alternatives denoted Scenarios 1, 2 and 3 were simulated and evaluated. Scenarios 1 and 3 were initiated by industrial process knowledge, whereas Scenario 2 was triggered by the outcome of activity A54. In Scenario 1, an improvement in operator training was assumed to halve the

p_k and t_k^{rem} of tasks that are highly affected by operator behavior—i.e., *B3*, *B4*, *B19*, *B21*, *C9*, *E7*, *E5*, and *F14*. These tasks are system sealing-related operations, such as leakage tests, as well as quality sampling, the success of which is influenced by the manual installation of the filling equipment. In Scenario 2, the substitution of an existing tank attachment with an automatic one was considered, and this would facilitate manual operations in block *D* and halve run times of this block. In Scenario 3, it was assumed that the blocks *G* and *I* could be removed from the process recipe of *post*-CIP/SIP. Usually, these blocks are necessary if the CIP/SIP is performed between two filling batches within a campaign—i.e., between *intra*-CIP/SIP processes. Because *post*-CIP/SIP is performed at the end of the campaign, after which the environment does not require sterility, this scenario was considered realistic.

Figure 4.8 shows the results of what-if analysis in *intra*-CIP/SIP and *post*-CIP/SIP on the left and right sides of the figure, respectively. The values of the probability distributions were normalized to the median $T^{\mathrm{CIP/SIP}}$ for the current (unmodified) process as the base scenario. In both processes, the outcomes of Scenario 1 did not differ from the base scenario. These results were analyzed by the Kolmogorov–Smirnov test with the null hypothesis that "the base scenario and Scenario 1 follow the same PDF" at a 5% significance level. The minimum p-values of the test were 0.17 and 0.19 for *intra*-CIP/SIP and *post*-CIP/SIP, respectively—i.e., the behavior of the operators would not preclude high process performance. In Scenario 2, a decrease of 12% in the mean total run time was found. Even if installing the new attachment would require testing and qualification, 25% of the total annual run time is invested in *intra*-CIP/SIP; therefore the potential annual run time reduction of 3% (25% times 12%) would encourage modification of the process. In Scenario 3, the measures were found to reduce the mean total run time by 40%. Based on this analysis, the modifications proposed in Scenario 3 were recently implemented at the investigated facility, and a 32% reduction in run time was observed in one commercial *post*-CIP/SIP.

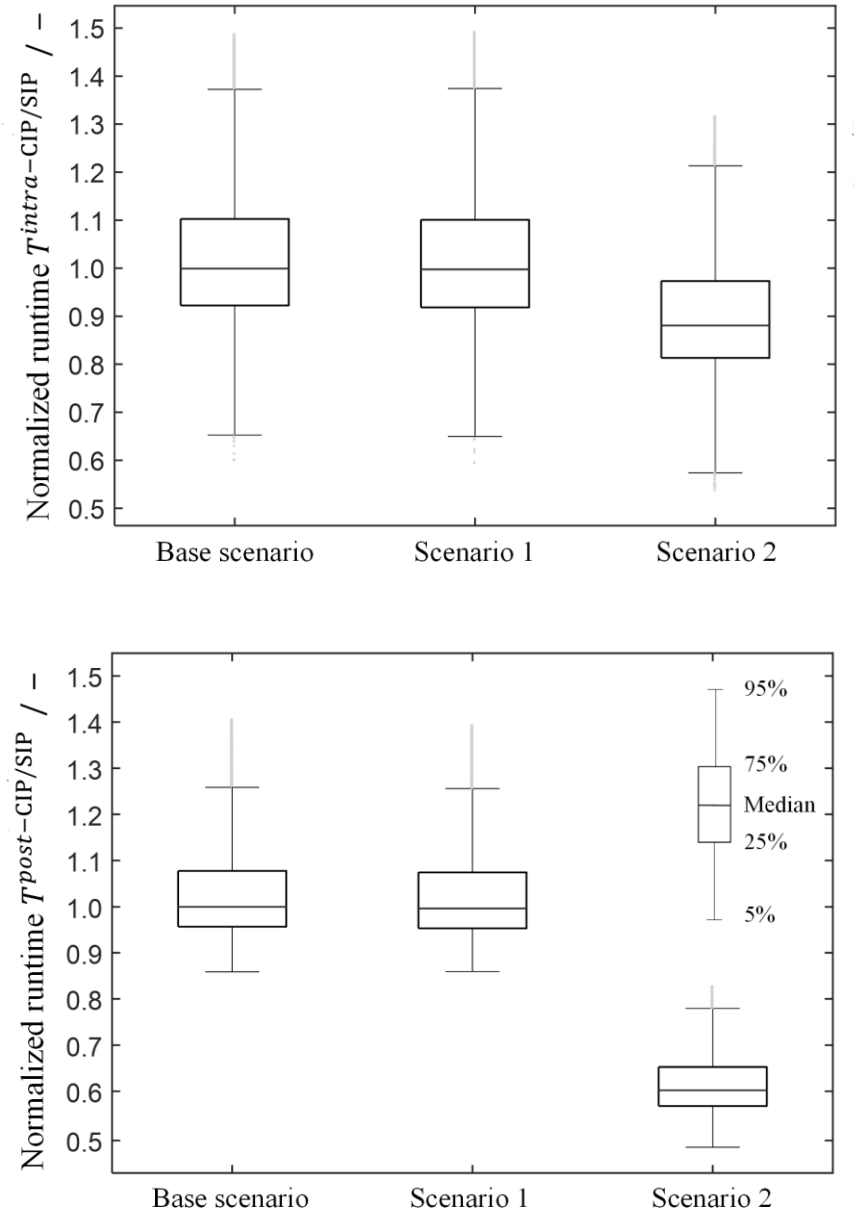


Figure 4.8 Scenario evaluation for *post*-CIP/SIP (top) and *intra*-CIP/SIP (bottom).

4.4 Results and discussion

In this section, the presented approach and the conventional LSS approach are compared using the case study. The result that would have been obtained by LSS, namely the runtime ratio, $ratio_k^{\mathrm{LSS}}$, was calculated to represent the annual contribution of task k to the total run time (see section B.4 LSS assessment in the appendix for detail).

Table 4.3 Summary of the outcome of the process performance assessment of CIP/SIP, comparing the presented approach with the conventional approach.

***intra*-CIP/SIP**						***post*-CIP/SIP**					
ID	Task description	$\bar{\rho}_k$	Φ_k	*RNE*	$ratio_k^{LSS}$	ID	Task description	$\bar{\rho}_k$	Φ_k	*RNE*	$ratio_k^{LSS}$
D2	Manual operation	0.94	0.85	low	0.23	*F18*	Rinsing	0.77	0.8	low	0.05
F18	Rinsing	0.52	0.8	low	0.03	*F5*	Manual operation	0.78	0.8	low	0.003
F5	Manual operation	0.60	0.8	low	0.01	*B5*	Pressure testing	0.53	0.6	high	0.02
						H19	Cooling	0.53	0.5	low	0.03

Table 4.3 presents a summary of the outcomes. It can be seen that task *F5* in *post*-CIP/SIP has a low $ratio_k^{\mathrm{LSS}}$ compared with *F18*; however, the coefficients $\bar{\rho}_k$ of the two tasks are very similar. Because the tasks *F5* and *F18* belong to the same process block that must be repeated in case of failure (see Eq. (4.11)), c_k^{rep} was the same for both tasks. In addition, the sampling pool U was similar for *F5* and *F18*, which led to the similarity in $\bar{\rho}_k$; however, the μ, and σ of the sampling pools were different. In fact, the value of $\mu_{F18}^{\mathrm{task}}$ was double that of μ_{F5}^{task}, and because *F5* was conducted manually, $\sigma_{F5}^{\mathrm{task}}$ was 15% higher than $\sigma_{F18}^{\mathrm{task}}$. The index $ratio_k^{\mathrm{LSS}}$ considers only the summation of the run time over batches and not the variation between batches, so the impact of *F5* was much lower than that of *F18*. Looking at the tasks *H19*, *B5*, and *F18* (which were automated), the results for the two approaches were similar. For these reasons, one could interpret the results as indicating that the conventional approach could underestimate the importance of manually operated tasks, which tend to have high process variability.

In addition to considering process variability, the results for the feasibility indicator Φ_k and RNE were obtained, as summarized in **Table 4.3**. The indicators Φ_k was assessed independently of the calculation of the PRCCs, and hence Φ_k could be incorporated in the LSS approach as well, if desired. The intention of RNE was to avoid overestimation of the PRCCs. Because a model-based approach is used for calculating RNE, it cannot

be directly compared with LSS. However, the presented methodology provided additional features that would increase the comprehensiveness of the assessment and deliver implementable suggestions for process modification.

The methodology has some potential limitations. First, excessive model complexity might exponentially increase the computational effort, which would render the MCS-based calculation of PRCCs unfeasible. Second, a lack of process knowledge would require the use of a metamodel or a black box model, which would remove the pharma-specific attributes and also the opportunity to evaluate alternative scenarios in activity A6. Third, in the presence of highly nonrobust processes, e.g., processes during the start-up phase, the performance model would show nonmonotonic behavior, rendering the PRCC calculation inappropriate. In such a situation, the use of alternative and more computationally demanding GSA methods, such as Sobol's index analysis,[59] would be required. Lastly, the methodology is suitable for single processes but does not assess the total performance of multiple and parallel processes that have shared utilities, such as WFI and workforce—i.e., factory-wide performance.

4.5 Conclusion

In this chapter, an uncertainty-conscious methodology was presented that can assess process performance and facilitate process improvement in biopharmaceutical DP manufacturing. The work is described as an activity model using IDEF0, which defines the information or tools needed to execute each activity. By executing the methodology, the tasks can be identified that most affect process performance, taking into account both nonroutine events (task failures) and process variability, the prominent characteristics of the process. The methodology also supports the consideration of industry-specific characteristics such as GMP requirements, the effort required to implement specific process modifications, and the risk of overestimating the improvement potential. The integrated approach serves as a solid basis for providing rational suggestions toward the realization of superior manufacturing processes for biopharmaceutical DPs.

There are three novelties in the presented work. First, process performance was formulated as a hybrid stochastic-deterministic model that can be used in a GSA with two nested loops. The outcome of the analysis—the PRCCs ρ_k—indicates the contribution of the design of each task to the total process performance under a given operational uncertainty. Second, the feasibility indicator Φ_k incorporates the knowledge of industrial

experts by reflecting manufacturing, economic, and strategic factors, to screen out task modifications that would highly affect process performance but are unlikely to be implemented because of low feasibility. Lastly, the evaluation of RNE acknowledges the needs and concerns of the industry about ensuring the heavy capital investment often required for process modifications is unlikely to be wasted.

An industrial case study was conducted to demonstrate the applicability of the methodology. Two types of cleaning and sterilization processes, *intra*-CIP/SIP (applied after each batch) and *post*-CIP/SIP (applied after each campaign), were analyzed with the aim of identifying the tasks to improve. After defining the overall process runtime as the overarching KPI, a process performance model that considered operational uncertainty was created and validated. A stochastic PRCC analysis was conducted, and it resulted in the quantification of the importance of each task for the overall process performance. The subsequent assessment of feasibility and RNE resulted in the identification of three tasks that could be improved in *intra*-CIP/SIP and *post*-CIP/SIP. The three scenarios analyzed in the what-if analysis suggested two process modifications—streamlining the process and redesigning a piece of equipment—that could reduce the mean total run time by 40% and 12%, respectively. The former has been implemented at the investigated facility, resulting in an actual reduction in the process runtime by of 3 h per campaign. Besides assessing the performance of the process considering the failures, the framework (see **Figure 2.1**) considers the reduction of downtimes caused by such failures through the prediction and prevention of unwanted events as it is presented in **Chapter 5**.

4.6 Nomenclature

B	Bernoulli PDF of failure in task k	–
$\boldsymbol{C}$	Inverted symmetric rank matrix	
$c_{k',k''}$	Element of the inverted symmetric rank matrix	–
c_k^{rep}	Failure counter for task k	–
ϵ	Normalized PRCC	–
$\boldsymbol{F}$	Failure matrix	
f_k	Boolean parameter representing success/failure in task k	–
$f_{k,j}$	Boolean parameter representing success/failure in task k in failure layer j	–

$\hat{f}$	PDFs of KPI_k and t_k^{op}	–
Φ_k	Feasibility indicator	–
j	Index of the failure layer, $j \in [1, +\infty]$	
K	Number of tasks in one process	–
$\boldsymbol{KPI}$	Manufacturing data matrix	
KPI	Total key performance indicator	–
KPI_k	Key performance indicator specific to task k	–
k	Task, $k \in \{A1, A2, \dots, J3\}$	
k'	Task, $k' \in \{A1, A2, \dots, J3\}$	
k''	Task, $k'' \in \{A1, A2, \dots, J3\}$	
M	Multinomial PDF of consecutive failures of task k	–
μ	Average rank	–
μ_k	Mean of the PDF $\hat{f}(KPI_k)$	h
N	Number of iterations for LHS–MCS	–
N_i	Number of iterations in the inner loop	–
N_o	Number of iterations in the outer loop	–
n	Term counter for the progression $u_n^{(k)}$, $n \in [0, K-k]$	
n_i	Iteration counter in the inner loop, $n_i \in [1, N_i]$	
n_o	Iteration counter in the outer loop, $n_o \in [1, N_o]$	
op	Operation type, $op \in \{\text{task}, \text{rem}, \text{corr}\}$	
p_k	Failure probability of task k	–
$p_{k,j}$	Failure probability of task k in failure layer j	–
q_k	General failure counter for task k	–
$\boldsymbol{R}$	GMP-defined repetition matrix for the overall process (related to tasks k', k)	
$\boldsymbol{R}^{\text{CIP/SIP}}$	GMP-defined repetition matrix for the CIP/SIP process (related to tasks k', k)	

$\boldsymbol{R}^{intra-\mathrm{CIP/SIP}}$	GMP-defined repetition matrix for the *intra*-CIP/SIP process (related to blocks X, Y)	
$\boldsymbol{R}^{post-\mathrm{CIP/SIP}}$	GMP-defined repetition matrix for the *post*-CIP/SIP process (related to blocks X, Y)	
$\boldsymbol{R}^{x}$	GMP-defined repetition matrix per process block x (related to tasks k', k)	
$Rep(k)$	Set containing the indices k' of the tasks repeated in the case of failure of task k	
$ratio_k^{\mathrm{LSS}}$	Ratio of the runtime of task k to the total run time	–
$r_{k,n}$	Rank of task k in sample n in the merged matrix [$\boldsymbol{X}$ $\boldsymbol{KPI}$]	–
ρ_k	PRCC of task k	–
ρ_{k,n_o}	PRCC of task k in sample n_o	–
$\overline{\rho}_k$	Mean of ρ_k	–
σ_k	Standard deviation of the PDF $\hat{f}(KPI_k)$	h
$T^{\mathrm{CIP/SIP}}$	CIP/SIP process runtime	h
$T_{n_i}^{\mathrm{CIP/SIP}}$	CIP/SIP process run time in sample n_i	h
$T_{CIP/SIP}^{\mathrm{annual}}$	Total annual run time invested in CIP/SIP	h
t_k	Student's t-test t-value of the PRCC of task k	–
t_k^{annual}	Total annual time invested in task k	h
t_k^{op}	Runtime of task k specific to operation op	–
U	Uniform PDF used in LHS–MCS	–
$u_n^{(k)}$	Geometric progression of the joint probability $p_{k,j}$	–
$\boldsymbol{X}$	Sample matrix	
X	Block that needs to be repeated, $X \in \{A, B, \ldots, J\}$	
x	Process block, $x \in \{A, B, \ldots, J\}$	
Y	Block related to interrupted task k, $Y \in \{A, B, \ldots, J\}$	

Chapter 5: Intelligent real-time prediction of imminent failures in the cleaning and sterilization process of biopharmaceutical manufacturing

(*Based on the manuscript in currently preparation by G. Casola, C. Siegmund, M. Mattern, and H. Sugiyama*)

5.1 Introduction

The rise of digitalization, in recent years, has drawn the attention of numerous industry sectors, and pharmaceutical companies, which are increasing their effort in the fourth industrial revolution. In a review article, Kemppainen (2017) presented the challenges and opportunities in transforming the pharmaceutical industry through digitalization.[89] The pharmaceutical industry operates under an extremely controlled environment that involves conservative regulations on quality for both process and product; it relies on quality of process and product by testing and QbD. Some products, such as injectables, are administered directly into the patient's body, herein, pharmaceutical companies cannot rely on a "statistically significant" quality but have to deliver a high-quality product at all times.[23] Process Analytical Technology (PAT) is well-accepted by the regulators, e.g., FDA, and is implemented in the control of the pharmaceutical process to assure QbD.[144,145]

Latent variable methods, such as Principal Component Analysis (PCA) are commonly applied techniques for chemometrics involved in PAT. Rajalahti and Kvalheim (2011) presented an overview on the use of the multivariate methods PCA and Partial Least Square (PLS) regression in a combination of characterization techniques, such as vibrational spectroscopy and imaging, for monitoring process and product characteristics in drugs manufacturing.[146] De Beer et al. (2009) presented a study on the use of PCA to analyze spectrometric data in the monitoring of a pharmaceutical freeze-drying process;[147] Wu and Khan (2010) developed an integrated PAT approach for the real-time monitoring of a pharmaceutical coprecipitation process, the trajectories of which were identified by applying PCA on infrared spectral data.[148] Kimura et al. (2014) applied PCA and multiple regression analysis for examining the correlation between operation conditions—e.g., granule and air flow rates—, material attributes—e.g., water content—, and micrometrics—e.g., granule size and density—in the production of granules by a multi-functional rotor granulator.[149] Chemometrics is commonly used in industrial practices; however, according to an industrial survey, other advanced statistical methods, which are mostly based on machine learning,[150] are yet not implemented because of their perceived complexity for non-experts.

Machine learning has been primarily used in pharmaceutics for drug discovery and pharmacovigilance. Previously in 2001, Burbidge et al. showed the application of Support Vector Machine (SVM) in the classification of new molecular compounds in the analysis of the structure-activity relationship.[151] In a review study, Hou et al. (2006) presented a list of analytical approaches, such as Neural Network (NN) and PLS, and

their use for predicting and classifying drug interaction characteristics such as absorption and permeability in the virtual screening of new molecules.[152] Garcia-Munoz and Mercado (2013) showed a latent-variable based model used in the optimal selection of raw materials for drug product design. The model that predicted the dissolution rate of the product from the raw material characteristics was optimized considering the availability of the excipients.[135] In a recent review paper, Gawehn et al. (2016) presented the advantages and limitations of various NN for supporting the modeling, discovery, and design of new drugs.[153] Zhao and Henriksson (2015) presented an application of the random forest learning algorithm for characterizing the weights of clinical events, e.g., side and collateral effects of drug consumption, during the clinical trials. Several additional studies were also published in recent year on similar topics.[154,155]

In addition to drug discovery and pharmacovigilance, the impact of such machine learning methods could be extended to the quality control of products and on the performance of processes. The current costly quality control on finished goods could be substituted by a product quality prediction and online monitoring. Similarly, the process performance can be improved by introducing predictive intervention policies—i.e., online fault detection, predictive maintenance. Regarding product quality, Gams et al. (2014) developed a machine learning-based method that incorporated commercial data and human decision to facilitate the manufacturing and assure the quality of tablets.[156] In a recent study, Akseli et al. (2017) proposed a machine learning tool for the prediction of tablet breaking force and disintegration, which are conventional indicators of a tablet's quality. The tool based on models, such as genetic algorithm, NN, and SVM, would enable the practical instauration of QbD by connecting material attributes with the quality indicators.[157] Contrary to other industrial sectors, in pharmaceutics, the implementation of preventive actions based on retrospective studies is more favorable because of the maturity of the reliability engineering methods. Some studies have been found to show the development of innovative predictive policies in various industrial sectors such as machining,[158,159] energy production,[160] and transportation.[161]

Predictive intervention policies have great potential in improving the performance of pharmaceutical processes. In fact, from a preliminary analysis on the case study on the potential performance bottlenecks presented in **Chapter 5**, the avoidance of scheduled (preventive) maintenance and the early detection of process failure could increase the time dedicated to production by roughly 20%. Regarding predictive maintenance

policies, Gao et al. (2015) reviewed the historical evolution of prognosis techniques in various fields with a focus on manufacturing systems; Gao et al. mentioned the importance of system architecture and symbiosis for enabling cloud manufacturing.[162] Susto et al. (2015) proposed a multiple classifier methodology for predictive maintenance. The methodology compared different performance tradeoffs by predicting multiple health-risk policy scenarios, to make the ideal maintenance decision.[163] Roy et al. (2014) applied multivariate statistical process monitoring for the continuous verification and detection of abnormal situations of CIP/SIP batches in DS synthesis. In the study, Roy et al. showed the potential of predictive monitoring in four case studies.[164] More studies on the development and implementation of machinery health prognosis and maintenance policy can be found elsewhere.[165]

Although numerous studies have been presented above, in the current situation, a tool is still required, which leverages both, the traditional PSE methods and the novel data "science" techniques, to provide a tailored decision-making approach for pharmaceutical manufacturing. The approach must enable the integration of the complex data structure, typical of the pharmaceutical industry, for predicting production patters, gaining insights on, and supporting the operations. The design of a process can rarely be changed, so the policy of "salvaging the salvageable" is achievable by reducing the downtime caused by process failures during the manufacturing operations. Conventional approaches, such as time series analysis and vibration analysis, used for the predictive maintenance of engines, or machining tools are not suited in this study because the data was not following a wave-like behavior. In the current situation, an intelligent tool is required that can learn from exclusively commercial manufacturing data and can support the real-time decision-making resulting based on the process monitoring is required.

In this chapter, an algorithm for predicting imminent process failures and supporting the decision of taking preventive actions by only leveraging manufacturing records is presented. The algorithm consists of two parts: an intelligent failure prediction tool and a risk-based decision-making tool. The monitoring tool is a self-training classification model that predicts the success of a batch; whereas, the decision-making tool evaluates the risk of taking preventive actions by comparing the positive and negative impacts of the intervention and the reliability of the prediction. A positive impact is intended as the time salvaged as a result of the early stop of a batch that is meant to fail; on the contrary, a negative impact is time lost by interrupting a batch that was meant to succeed.

By integrating a self-learning algorithm, the intelligent monitoring tool re-trains itself based on its age or the evolution of its prediction performance using historical data.

The applicability of the algorithm is elucidated in an industrial case study, where the data historians of a CIP/SIP process performed in a sterile filling plant of F. Hoffmann-La Roche are leveraged. In the case study, the algorithm is used to predict imminent failures during a CIP/SIP batch, so avoiding the repetition of tasks and the resulting downtime. The goal of applying the algorithm is to salvage time by providing a decision to interrupt batches that are meant to fail before encountering the failure. The algorithm weighs the probability of failure and the risk of interrupting batches that are meant to succeed.

5.2 Algorithm

The algorithm shown in **Figure 5.1** is used for predicting imminent failures in the manufacturing in real-time. The algorithm is employed at time intervals Δt, hereafter named as the test time intervals. These intervals are defined accordingly to the target the resolution of the failure prediction—e.g., every 10 s, every 1 min, or every 1 h. The algorithm consists of two main parts: the failure prediction (see orange area, **Figure 5.1**) and the decision making (see blue area, **Figure 5.1**). In the first part, the dataset at time t_n is used to predict the failure class C—i.e., *successful* and *about to fail*—of the running batch at time t_{n+1}; in the second part, the prediction is used to suggest an action—i.e., *interrupt* or *do not interrupt* the batch—to be taken. The failure class of the batch at time t_{n+1} is predicted from the data recorded at time t_n; the failure class is a categorical variable, which classifies the batches by their future status—i.e., successful or failed. A classification model, which has been trained at time τ_i on data historian, is employed to predict the class C.

The failure class defined as *about to fail* suggests the presence of Predictable Failures (PF) at time $t > t_{n+1}$. If a PF is not detected no action is taken and the algorithm is iterated at time t_{n+1}; whereas if a PF is predicted at t_{n+1} the action team—i.e., operators, who are responsible for the batch—is put on standby, the production manager is informed, and the second part of the algorithm is initiated. A quantitative risk analysis on the impact of interrupting the batch and performing preventive maintenance, or continuing the batch until the $n+1$ iteration supports the decision-making. If a failure occurs after the decision of not stopping the batch, a

corrective action is required before restarting the batch. In the next sections, the two activities are represented in more details.

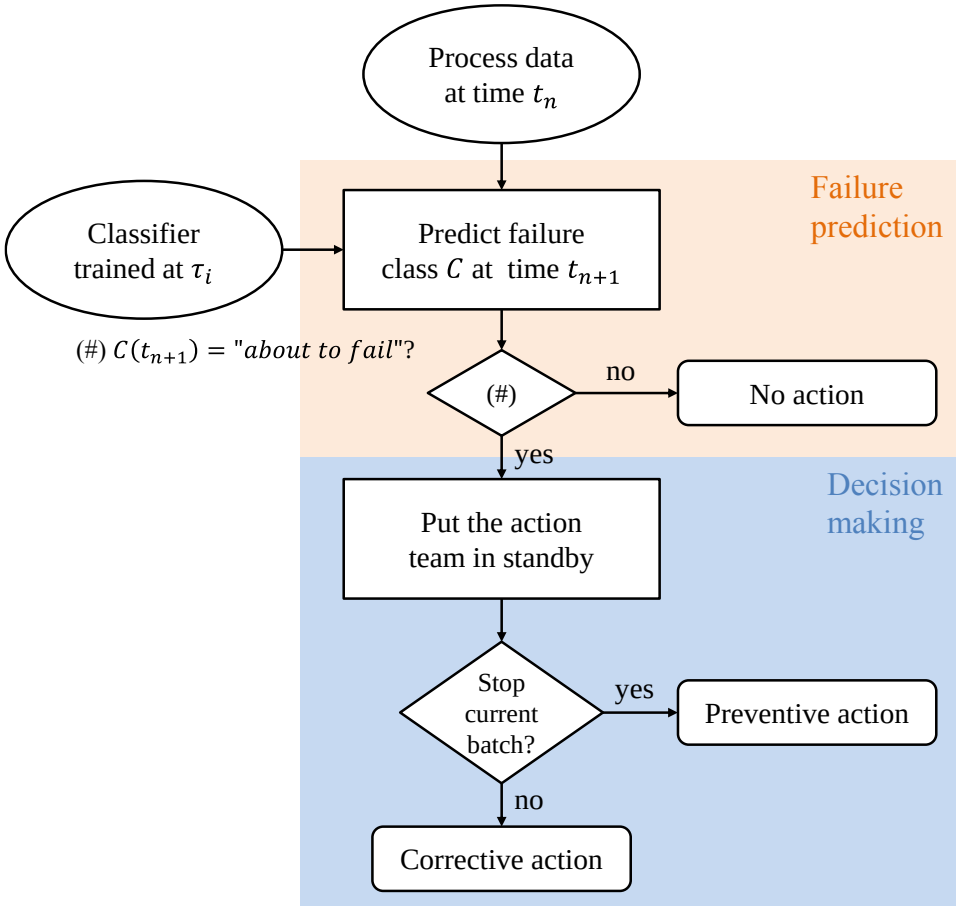


Figure 5.1 Real-time failure prediction algorithm.

5.2.1 Real-time failure prediction

The failure prediction model takes, as input, sensors data recorded from the current batch at time t_n to predict the failure classes, *successful* and *about to fail* at time t_{n+1}. As mentioned in the introduction, a large pallet of supervised models—e.g., DT, SVM, and k-Nearest Neighbors (KNN)—can be employed in this classification problem; however, because the model selection is not the core part of this work, a simple DT model was employed. In contrast to other predictive monitoring techniques, which predicted the process outcomes for specific events,[164] the machine learning model presented here is trained to answer to the question: "will the current batch fail in the imminent future?" with a "yes" or "no

The prediction of failure is introduced to reduce the downtime caused by unexpected failures during the operation; hence, the worst outcome after the implementation would remain the non-detectability of the failure (Type I error, see Appendix C.1). In addition, the model should avoid Type II errors, namely classifying prospectively *successful* batches as *about to fail*. To guarantee such a condition, and because accuracy—i.e., the number of true predictions divided by the total number of predictions—can not be used to quantify the quality of the classification, a new prediction performance indicator was introduced in Eq. (5.1).

$$NPV = \frac{true\ negative\ prediction}{negative\ prediction} \tag{5.1}$$

Negative Predictive Value (NPV) quantifies from 0 to 1 the quality of the model when predicting failures; a value of 1 guarantees a fully conservative prediction, namely it guarantees the absence of Type II errors. Having a model with high NPV is particularly crucial when dealing with unbalanced—i.e., only a few samples contain data of batches that are about to fail—and large-sized datasets—i.e., big data size—; the conventional accuracy indicator does not deliver a significant quantification of the model prediction quality.

5.2.1.1 Model training

The decision tree classifier is trained, validated, and tested with the labeled training data. The size and variability of the input data set are usually remarkably large for manufacturing sensor data. In the case study presented in section 5.3, which represents a small end of data variety in the industry, the number of data points (time series for multiple sensors) sums up to a total of 1.2 billion. Various workflows have been published to support the transformation and adaptation of data;[166–168] however, in general, the choice of workflow depends on the morphology and quality of the data, on the purpose of the model, and on the process to which the model is applied. In this work, the data transformation workflow was defined as follow:

I. Knowledge-based data splitting (following the process recipe)
II. Scaling[169]
III. Dimensionality reduction[168]
IV. Size reduction (if necessary)
V. Relabeling
VI. Balancing[170] (if necessary)

As mentioned in the introduction the algorithm aims at assembly-like processes that follow a specific recipe, therefore sensor values are process-task and time dependent. The training dataset is split into specific datasets, which are used to train block—i.e., group of tasks—specific classifiers (I). If the separation of the dataset by blocks does not eliminate non-causal correlations, the datasets are further divided into task-specific training dataset. The samples represented by x_i, where the index i defines the source of the data—i.e., the sensor from which the data point is originated—are scaled (II) following Eq. (5.2).[169]

$$x_i' = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)} \tag{5.2}$$

The large training and validation datasets are transformed with principal component analysis (PCA) to reduce the dimensionality (III); by reducing the dimensionality, the sample size shrinks automatically. Hereafter, principal components are noted as $P \in \{p_1, p_2, \dots, p_I\}$ where I is the total number of data sources. The size dataset can be further reduced (IV)—e.g., through random sampling—if the machine (computer) encounters limitations in handling very large information.

The process follows a recipe, which means that ambiguous data points—i.e., data points with the same *P* but different labels—are not determinant in estimating the class decision boundary; ambiguous data points lead to class overlap. The initial labeling is exported automatically from the data records, namely all data belonging to an interrupted or to a successful batch are labeled as *about to fail* or *successful*, respectively. To reduce class overlap and improve model reliability, the sample is relabeled (VI); ambiguous points are relabeled as *successful*. A more in-depth study on the relabeling of the training dataset is presented in section 5.3.1.2. Finally, if needed to further improve the model training, the sample is artificially balanced (V), e.g., through random up- or down-sampling of specific classes[170], to avoid yielding unprecise—i.e., low NPV—models.

5.2.1.2 Model maintenance

Batch processes, especially those located in a continuous performance improvement environment like pharmaceutical processes and processes that are influenced by human behavior, are under constant change. In the presence of evolving processes, the prediction model must evolve consequently; as mentioned the model is applicable only for data points located inside the training dataset. Thus a model maintenance strategy is required.

The classifier requires re-training on a scheduled basis because of two main factors. First, machine-learning models can only predict what they “learned”; the model is trained on historical data records exclusively, meaning that predictor data that are not contained in the training dataset boundary cannot be used for predictions. Second, commercial processes usually have control boundaries that allow the generation of “new” data over time. These two factors have to be incorporated into an intelligent and evolving algorithm that recognizes under which dataset conditions—i.e., known data space that delivers reliable predictions— under which it can be applied and when it requires retraining.

Figure 5.2 presents an intelligent model maintenance algorithm, which is used for the time or event-based retraining and learning of the model.

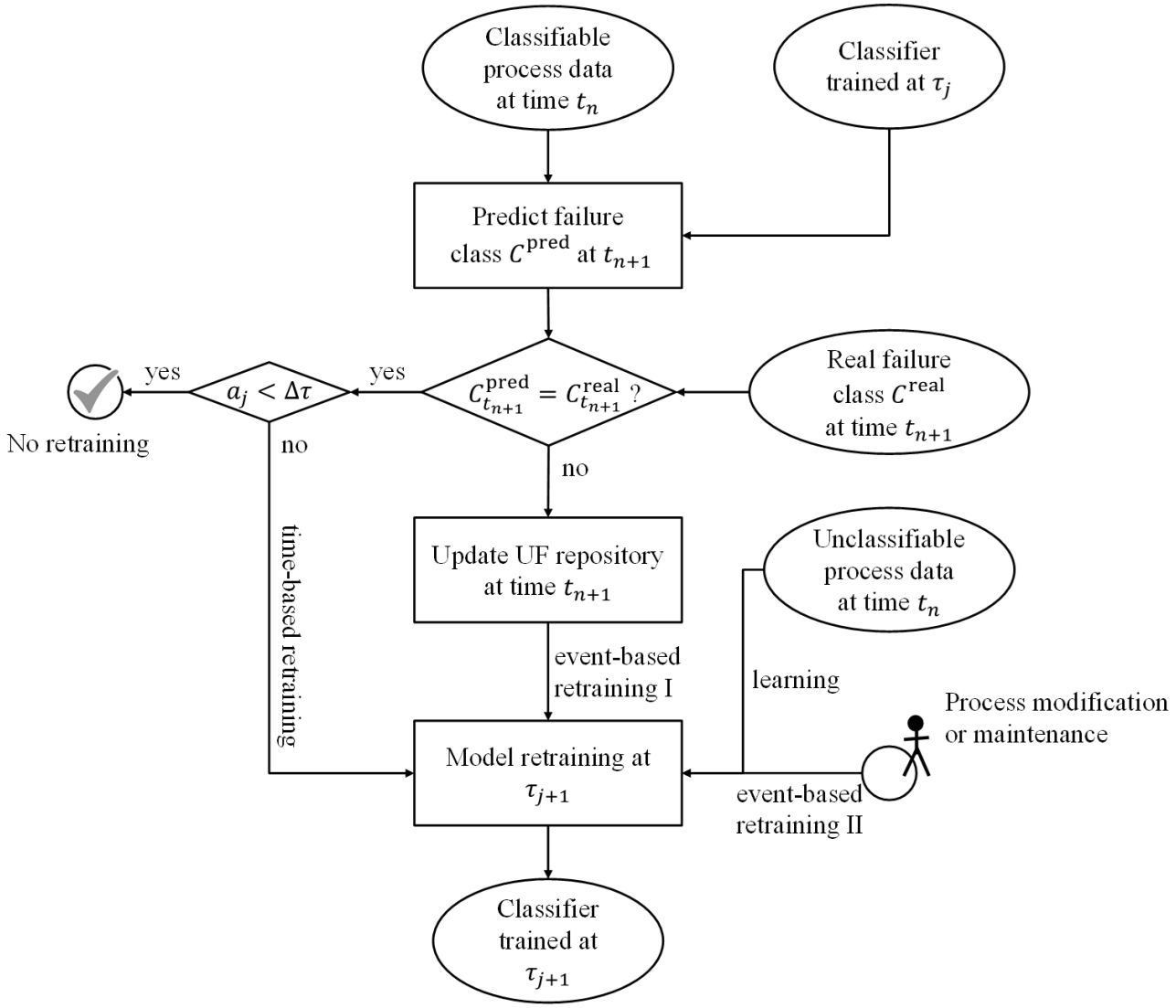


Figure 5.2 Intelligent classifier maintenance algorithm.

In **Figure 5.2**, τ_j represents the time at which the classifier was trained; the age of the classifier a_j is defined by Eq. (5.3).

$$a_j = t_n - \tau_j \qquad \text{where } a_j \leq \Delta\tau \tag{5.3}$$

Time-based model retraining is the most straightforward retraining strategy and is in vigor with the condition shown in in **Figure 5.2**. The age of the model a_j cannot become higher than $\Delta\tau$, which is defined according to the evolution status of the process, namely, start-up, growing, or stable status.

Event-based model retraining is required mainly for four reasons; the first reason is the misclassification of classifiable data, namely, a type I prediction error. In such an event the process breaks down without being predicted. The error is detected at time t_{n+1} when the predicted failure class $C_{t_{n+1}}^{\text{pred}}$ is different from the real failure class $C_{t_{n+1}}^{\text{real}}$ (see **Figure 5.2**). Such a failure is referred to as Unpredictable Failure (UF), the data of which are saved in a separated repository, the UF repository. The UF data is summoned from the UF repository and is used to retrain the classifier (event-based retraining I, **Figure 5.2**). Type II prediction errors cannot be identified once the algorithm is operative because the batches classified as *about to fail* are stopped before a failure occurs; therefore, it is important to apply the classifier only on data point combinations—e.g., principal component areas—located within the training dataset boundaries. The second reason is the inclusion of unclassifiable data—i.e., data points located outside of the training dataset of the trained model—in the new training dataset; which in this study is referred to as *learning* (see **Figure 5.2**). In presence of a big amount of data, as in this study, learning and event-based retraining I occur frequently and would require great attention. Whereas in this case, since the historical data is automatically labeled, the retraining of the model can be automated. The third reason is a substantial change in the process and equipment, e.g., new pumps, sensors, and operation parameters; in such an event, an external input and new data will be required (event-based retraining II, **Figure 5.2**). The last reason is plant maintenance, which is a less frequent but invasive intervention, such as sensors recalibration and gaskets installion. Plant maintenance can change the behaviour of the data (event-based retraining II, **Figure 5.2**).

5.2.2 Real-time decision-making

A tool was developed, which is user-friendly and uncomplicated to operate, for supporting the decision in interrupting the batch if a PF is detected. **Figure 5.3** shows the failure class monitoring plot (above) and risk analysis plot (below). The evolution of the failure class of each batch (solid line in **Figure 5.3**, above) is monitored through model failure prediction; the class C at time t_{n+1} is predicted from the principal component tuple $P \in \{p_1, p_2, \dots, p_I\}$ at time t_n (only p_1 and p_2 are shown in the graphical representation). The decision-making is performed with the two approaches; the first (see **Figure 5.3**, below) and most conservative is an adaptation of a risk analysis plot—i.e., frequency vs. impact, where risk, $Risk$, is defined by Eq. (5.4).

$$Risk = Im \cdot (1 - NPV) \tag{5.4}$$

The event frequency, which is used in the conventional risk description, is substituted by the NPV; the impact Im is the time loss and cost for the restarting. The risk considered in this decision-making process is the risk of worsening the process performance, namely the duration of the batch, through a preventive intervention, and the impact Im is defined in Eq. (5.5), where t_0 is the batch starting time.

$$Im = t_n - t_0 \tag{5.5}$$

The impact is defined as the time already invested in the process, which is the time that would have to be reinvested in the case of an intervention. The second approach is the evaluation of the expected value of the time gain G in case of an intervention defined in Eq. (5.6) and (5.7)

$$Im = \mathrm{E}[G] = \hat{g} \cdot NPV - Im \cdot (1 - NPV) \tag{5.6}$$

$$\hat{g} = \hat{t}_{\mathrm{failure}} - t_n \tag{5.7}$$

The variables $\mathrm{E}[G]$, $\hat{g}$, and $\hat{t}_{\mathrm{failure}}$ are the expected value of the time gain, the estimated time gain for the local decision and the estimated time at which the failure will occur; the variable $\hat{t}_{\mathrm{failure}}$ is estimated from historical failures.

Figure 5.3 shows a graphical example of the tool in general terms: the evolution of the failure class is monitored; if the profile remains inside the successful, e.g., Batch 1, no intervention is required, whereas if the profile crosses the decision boundary, as it is shown for Batch 2, a risk evaluation is initiated as mentioned in **Figure 5.1**. While the risk is considered acceptable (orange and blue cross, **Figure 5.3**) no intervention is triggered; however, if a risk is considered unacceptable, the batch is interrupted (green cross, **Figure 5.3**).

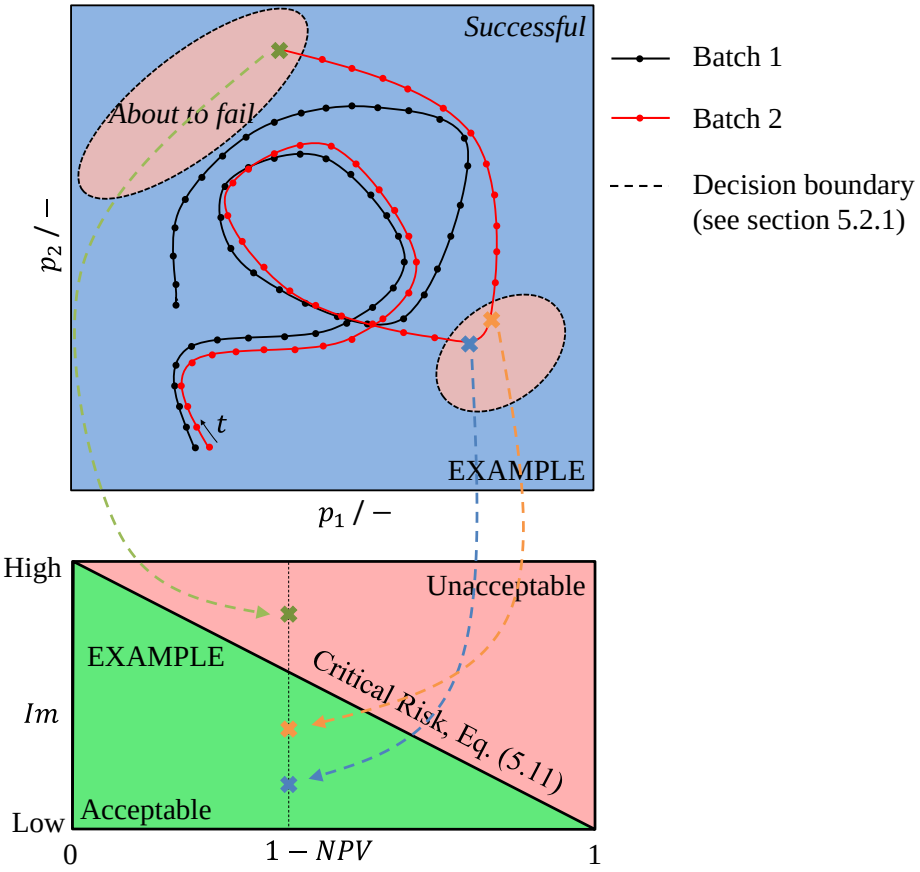


Figure 5.3 Decision-making tool.

5.3 Case Study

The algorithm shown in **Figure 5.1** was applied to the data records produced by a cleaning and sterilization process of F. Hoffmann–La Roche Ltd. in Switzerland. The CIP/SIP process is responsible for the cleaning and sterilizing of the product contacting surfaces of a filling plant for sterile biopharmaceutical drug solution in vial form. The process is batch and follows the recipe, which sets the boundaries for the process parameters—i.e., pressure temperature, and duration—to be applied to each task to run the process successfully. **Table 5.1** shows the recipe of the pre-campaign CIP/SIP process, which consists of 252 tasks divided into 11 blocks.

Table 5.1 Process recipe of the *pre*-CIP/SIP.

ID	Block description
A	Preparation of the CIP/SIP
B	Rinsing the piping with DW
C	Filter integrity test
D	Impermeability testing of the filling needles
E	Decontamination of the isolator
F	Rinsing the piping with WFI
G	Sterilization of the system
H	Drying and cooling of the piping
I	Integrity testing of the production filter after SIP
J	Integrity testing of the production gas filter after SIP
K	End of CIP/SIP

Numerous valves present in the plant regulate the pressure in the piping; however, the facility does not have any active system—i.e., model predictive control or PID controllers—as a control. The process parameters are constrained by upper and lower control boundaries for each sensor—i.e., temperature and pressure—for each task. Whenever a signal crosses a control boundary, the process stops. Because of the absence of an active controlling system the process parameters can vary within the control boundaries suggesting that it is plausible to hypothesize that particular variation can lead to process failure. The monitored data of the process consisted of time, five pressure profiles and nine temperature profiles, which recorded $I = 15$ signals at 1 Hz frequency for 4 years providing the data of 238 batches; the metadata—i.e., information on batch and block ID, and time—was exported for supporting the data preprocessing. The physical sensor data was filtered by time using the batch cluster boundaries (see **Figure 3.13**), namely starting and ending times of each batch, which was the outcome of the algorithm in **Chapter 3**. A simplified graphical representation of the filling system is shown in **Figure 1.7**, but the sensors position could be provided because of the high complexity of the piping network.

5.3.1 Real-time failure prediction

5.3.1.1 Data transformation

The labeled data of 238 batches was imported, and was further transformed following the workflow presented in Section 5.2.1.1. First, the metadata were exploited to sort and group the data by block (I). Although the process blocks are independent from each other and have different process requirements, their data points overlapped. It was possible to avoid non-causal correlations by dividing the dataset and creating one classifier per block; the resulting classification model consisted of 11 independent block-specific classifiers. Second, all data points (including the time) were scaled (II) using Eq. (5.1); second, after the PCA (III), the first six principal components, which were describing 95% of the variability, were selected as the new process features $\{p_1, p_2, p_3, p_4, p_5, p_6\}$. For the sake of readability only p_1 and p_2 (77% of the variability) are represented graphically; however, all the models and decisions take in consideration all the six components (IV). Fourth, the main dataset was split in two datasets, namely the training and deployment datasets, containing the data of 222 and 16 batches, respectively Half of the batches in the deployment set were chosen from the successful batches and the other half from failed batches (where one or more operating steps had to be repeated). Fifth, the training dataset was relabeled to reduce the degree of overlap[171] (DOO) between the classes (V); a detailed study on the effect of DOO is found later in the thesis. Last, the training dataset was balanced by up-sampling (VI) the data belonging to batches labeled as *about to fail*.

5.3.1.2 Relabeling

The relabeling of data points labeled as *about to fail* into *successful* in high-DOO areas (hypervolume if all six components are visualized) is plausible, whereas the contrary is not; the relabeling is valid under the assumption that a process is successful until it fails. The DOO between the two classes is defined using fuzzy sets as it is shown in Eq. (5.8) and Eq. (5.9)

$$d^+|_V = \frac{\#\text{points (with } C = about\ to\ fail) \text{ in } V}{M|_V} \tag{5.8}$$

$$\begin{cases} DOO|_V = 2 \cdot d^+|_V & if\ d^+|_V < 0.5 \\ DOO|_V = 2 - 2 \cdot d^+|_V & if\ d^+|_V \geq\ 0.5 \\ DOO|_V = 0 & if\ M|_A = 0 \end{cases} \tag{5.9}$$

The parameters $d^+|_V$ and M are the density of the points labeled as *about to fail*, and the total number of data points located inside the hyper volume V, respectively. The volume V is a local volume inside the process space. Full class overlap ($DOO = 1$) is observed when the number of points inside a local volume V is the same for both classes. Two factors influence the relabeling activity: the size of the local volume and the highest level of data ambiguity that is accepted prior to the model training. The size of the local volume is a function of the granularity parameter γ, which determines the number of local volumes that are found on each dimension of the process space and are used for the calculation of the $DOO|_V$. The calculation of the local volume V is shown by Eq. (5.10); each principal component dimension is divided into segments, the number of which is described by the parameter γ.

$$V = \gamma^{-I} \prod_{i}^{I} [\max(p_i) - \min(p_i)] \tag{5.10}$$

The granularity determines the resolution of the relabeling and can be an essential parameter in the determination of the capability of the classifier to represent the reality. The second factor is the critical degree of overlap DOO_c. The DOO_c determines the critical overlapping degree allowed inside a local volume; if the condition $DOO|_V < DOO_c$ is valid, relabeling is not required. The effect of the two parameters on the quality of the classification is shown in the next section by sensitivity analysis (see **Figure 5.5**).

The relabeling task reduces the ambiguity in data and has the effect of sharpening the boundaries between the failure classes. A practical demonstration of the relabeling step is shown for the data of the process block *C* ($\gamma = 10$, $DOO_c = 0$) in **Figure 5.4**, where the data points labeled as *about the fail* (red dots) located in areas (due to the 2D representation) with high overlap are relabeled as *successful* (blue dots).

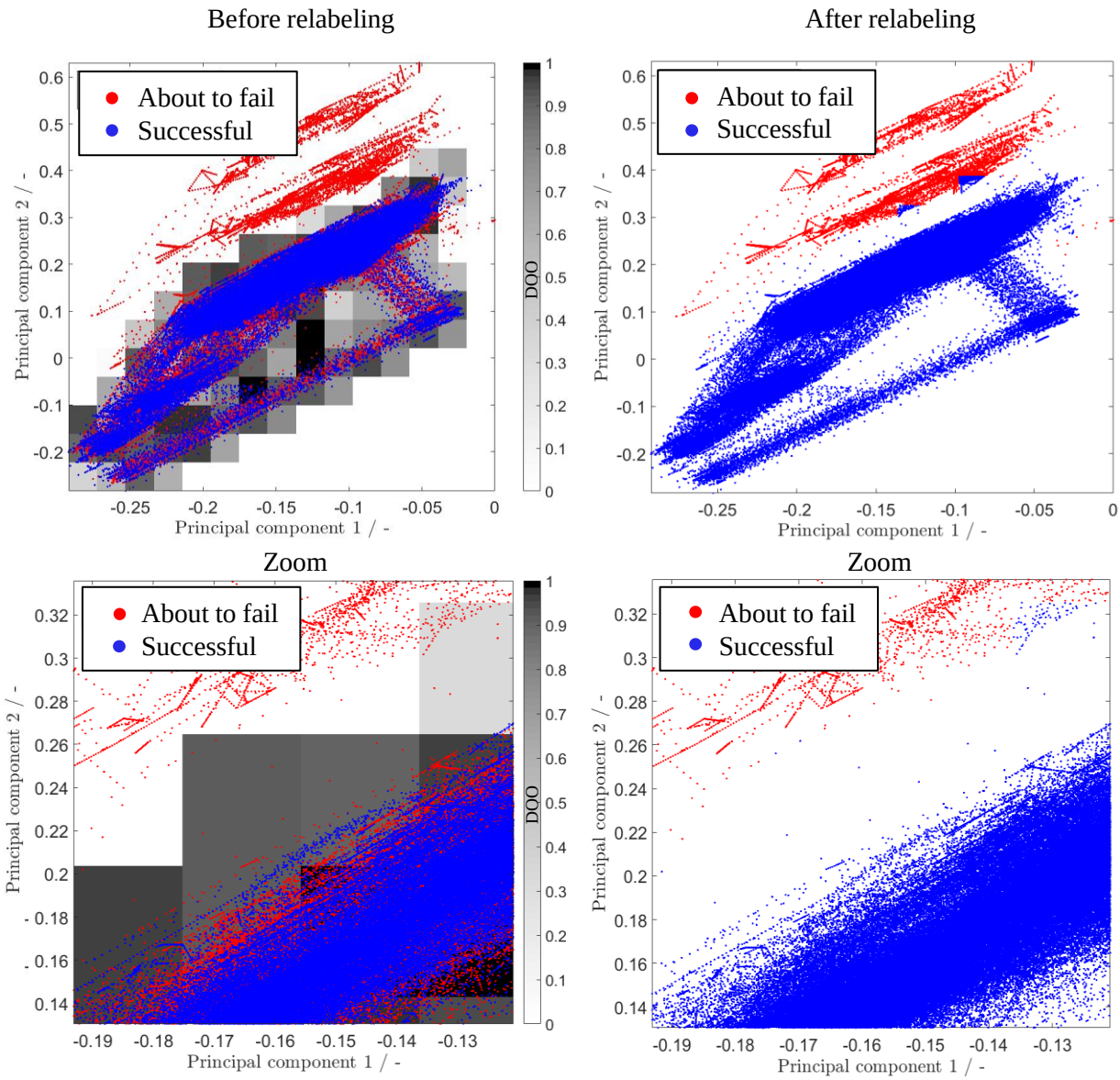


Figure 5.4 Plot showing data overlapping and outliers for an explanatory purpose.

Figure 5.4 shows the data landscape before (top, left) and the after (top, right) the relabeling of the data in the case of block *C*. Zoomed representations are shown in the bottom graphs of **Figure 5.4** for better visualization of the change in the labeling. The local degree of overlap, $DOO|_V$, was calculated and the local volumes with $DOO > DOO_c$ were relabeled; setting $DOO_c = 0$ is the most conservative approach because it removes the ambiguity form the training data and lowers the risk of type II classification errors. After the relabeling and the training of the classifier, the two classes are separated and a clear decision boundary can be drawn (see Section 5.3.1.4).

5.3.1.3 Model training

A preliminary analysis of the effect of the two parameters, namely γ and DOO_c, was performed before the training of the classifier used in the deployment. The parameters DOO_c (*x*-axis, **Figure 5.5**) and of γ (*y*-axis, **Figure 5.5**) were varied to identify the parameter that would lead to the most precise classification of the failure classes, namely, no Type II errors. For the sake of simplicity, the 11 sub-classifiers were trained with the same γ, DOO_c for each combination of the changeable parameters. **Figure 5.5** shows the results of the sensitivity analysis on the classifier performance, here represented by the mean *NPV* of the 11 blocks, for two different cases of process maturity. The maturity of a process/model was represented by the size of available data; the higher the number of batches included in the dataset, the more mature the process/model is considered. The two decision tree classifiers were multiply trained with 85 (**Figure 5.5**, left) and 238 (**Figure 5.5**, right) batches to evaluate the influence of the parameters, γ and DOO_c, on classification. The performance of the classifier was calculated by holding out 25% of the training data points during the training activity; this dataset, which was randomly sampled, was compared with the model predictions and used to calculate the NPV. The block-specific classifier sensitivity analysis are shown for the two maturity levels in **Figure C.2.1-C.2.2** in the Appendix.

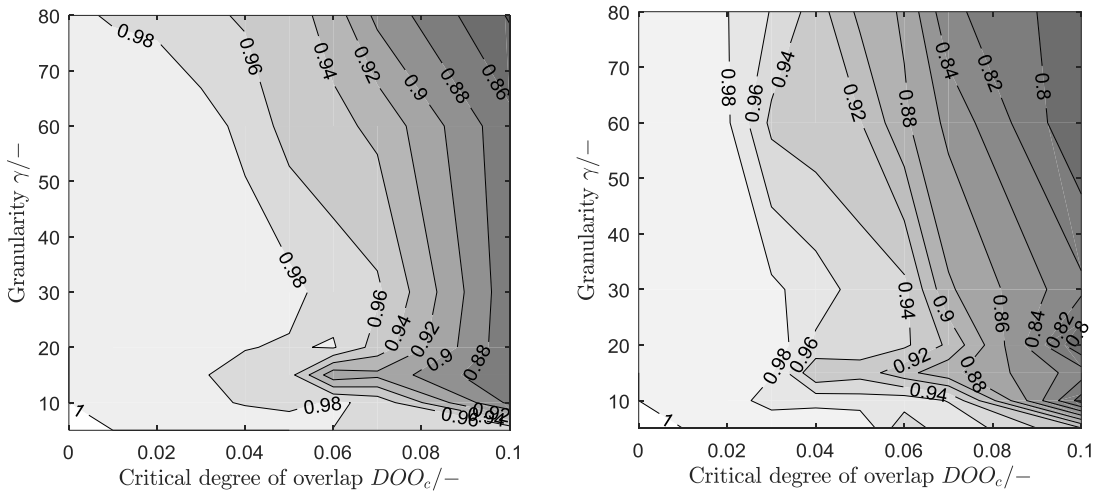


Figure 5.5 Parameter sensitivity analysis on the prediction performance, $\overline{NPV}$, for two maturity levels of the training datasets, namely 85 batches (right) and 238 batches (left).

Figure 5.5 show that the maturity levels does not play a significant difference on the selection of the training parameters. Two scenarios were defined; the first was named *conservative* and the second *risky*. The two sensitivity analyses do not differ; therefore, the parameters defining the scenarios were defined based on the

sensitivity of the mature model. In the first scenario, a conservative setting, which is best represented by a high NPV—i.e., $NPV = 1$—was selected; the combination of parameters $\gamma = 15$, $DOO_c = 0$ was used. In the second scenario, the parameter combination $\gamma = 15$, $DOO_c = 0.07$ was defined as *risky*—i.e., $NPV < 1$. The two scenarios are used to highlight the financial impact of the algorithm from the two strategical viewpoints.

It has to be mentioned that high granularity increases the computational time required in the relabeling step, which could limit the retraining activity for less mature classifiers that frequently receive new information. Low granularity, e.g., $\gamma = 5$, delivers great classification performance; however, setting low γ would result in a excessive relabeling of data points originally labeled as *about to fail*, increasing the number of Type I errors and reducing the specificity of the classification. This effect is notable in **Figures C.2.1-C.2.2**, in which the NPV could not be calculated in low granularity areas because of the absence of *about to fail* data points.

The two models, *conservative* and *risky*, were finally trained on four years of data (training dataset, 68 successful and 154 failed batches); the resulting mean NPV was equal to 1 and 0.81, respectively. The integral information regarding the performance of each single block-specific classifier (**Table C.3.1**) and the graphical representation of the decision boundaries (**Figure C.3.1-C.3.2**) for both models are shown in **Section C.3** in the Appendix.

5.3.1.4 Model deployment

After training the DT classifiers, the decision boundary between the two failure classes was drawn, and the model was deployed, as it is shown for the two scenarios in **Figure 5.6** and **Figure 5.7**, respectively.

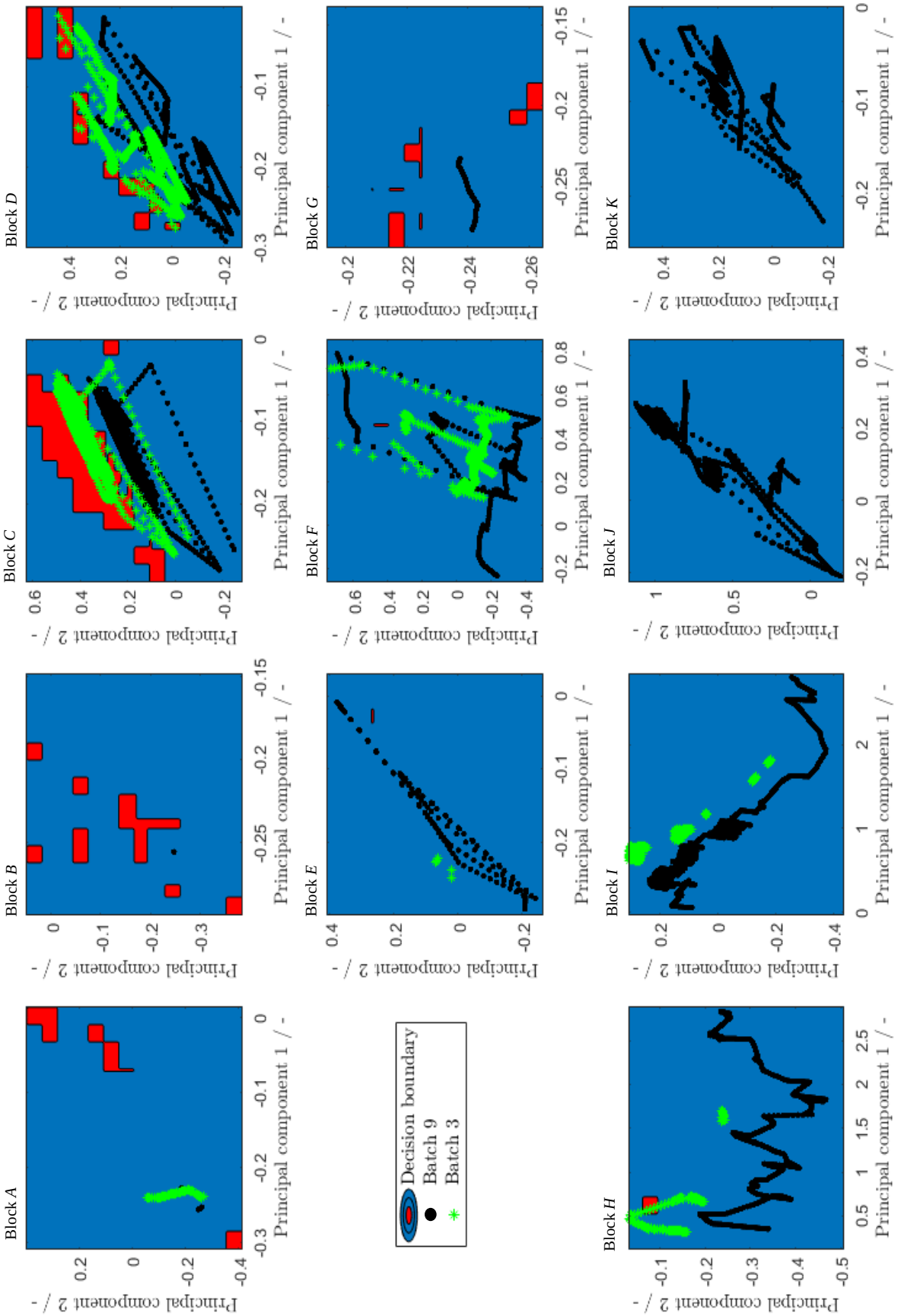


Figure 5.6 Real-time prediction of the failure class for the conservative scenario.

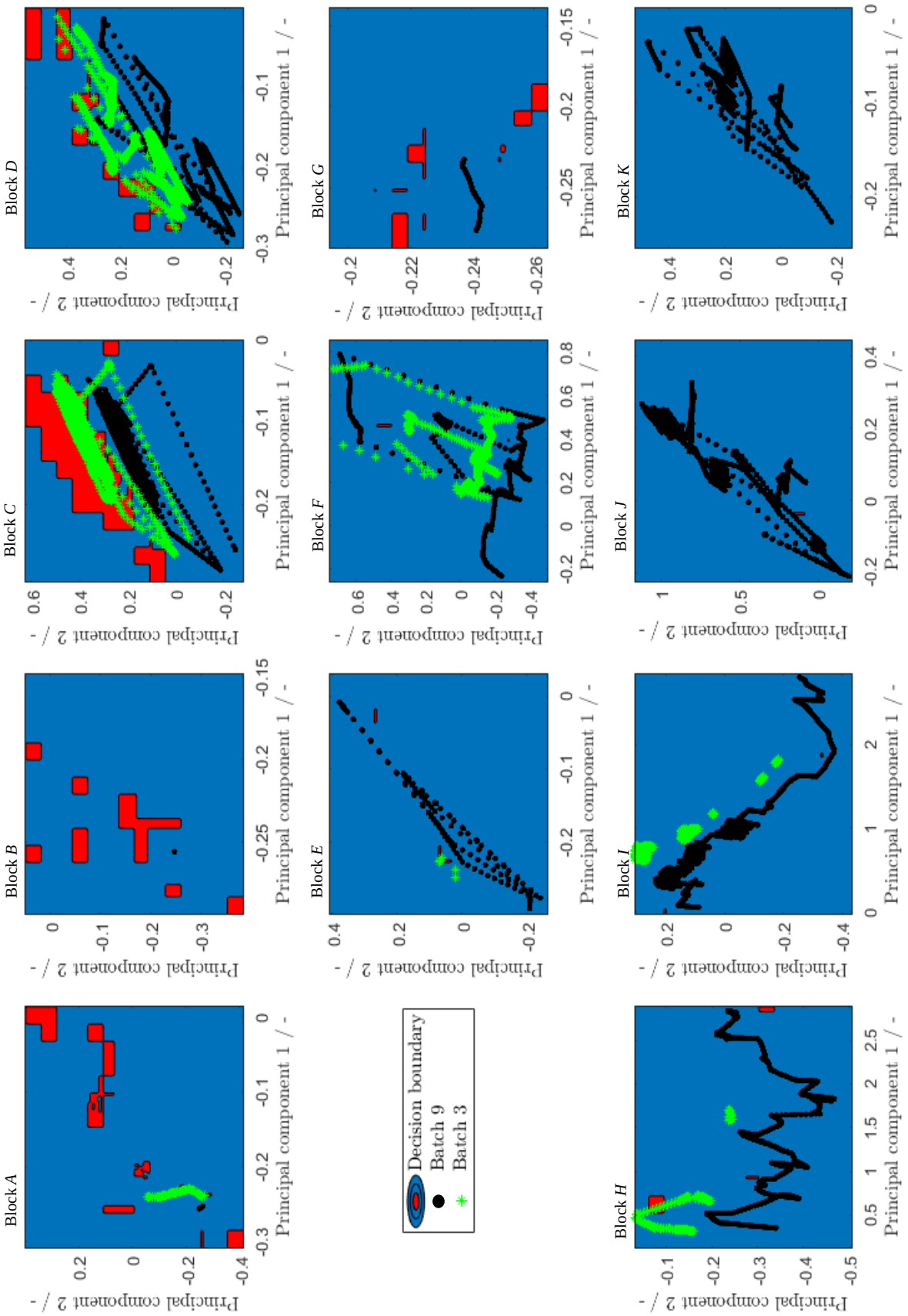


Figure 5.7 Real-time prediction of the failure class for the risky scenario.

In **Figure 5.6** and **Figure 5.7**, the process space colored in blue represents the *successful* area, whereas the red area represented the process space in which a batch can be classified as *about to fail*. The effect of having a model with an NPV smaller than 1 results in the increase of the areas where data are classified as *about to fail*. The models previously trained for the two scenarios were used to predict the status of 16 batches (deployment dataset) to test the applicability of the method to real operations. **Figure 5.6** and **Figure 5.7** show the data of a successful batch (Batch 9) and a failed batch (Batch 3) selected from the deployment dataset. The figures show that processes can be monitored and failure can be predicted in real-time (green profile). The two scenarios delivered slightly different decision boundaries. In the *risky* scenario, the model identified Batch 3 as failing already at its first block (Block *A*, **Figure 5.7**). On the contrary, in the *conservative* scenario the failure was detected only in block *C*. The repercussion of the delayed detection lead to the loss of 0.82 h (see **Table 5.2**) Real-time failure prediction was performed, and the results are shown in **Figure 5.8**

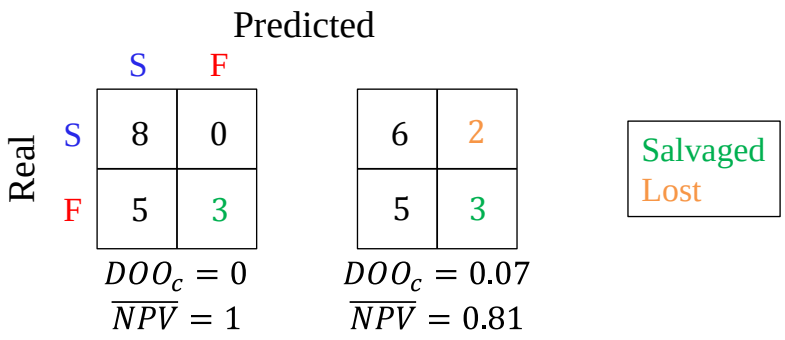


Figure 5.8 Real-time prediction of the failure class.

The result of the prediction and the quality of the classifiers, which are listed in **Table 5.2** are used in the real decision-making to evaluate the risk of taking preventive action. The time of detection is defined as the time as the first prediction equal to *about to fail*

Table 5.2 Result of the deployment of the classifier on 16 batches.

	Time of failure detection [h]			
Batch ID	*Conservative*	*Risky*	Status in reality	Real duration [h]
Batch 1	-		Failed	8.00
Batch 2	-		Failed	2.67
Batch 3	0.82	0	Failed	45.97
Batch 4	-		Failed	3.96
Batch 5	-		Failed	7.26
Batch 6	-		Failed	1.36
Batch 7	0.43	0.43	Failed	29.09
Batch 8	0.3	0.3	Failed	32.50
Batch 9	-		Successful	12.25
Batch 10	-	2.57	Successful	54.93
Batch 11	-	3.08	Successful	14.25
Batch 12	-	-	Successful	14.04
Batch 13	-	-	Successful	29.41
Batch 14	-	-	Successful	14.33
Batch 15	-	-	Successful	15.05
Batch 16	-	-	Successful	16.35

5.3.2 Real-time decision-making

Quantitative support of the decision-making on the intervention policy was provided by defining the critical risk in the impact-frequency approach. Gathering the information to clearly define the critical risk boundary from the business point of view is extremely challenging in an industrial setup. Experimentation on the use of the decision-making tool in real life is required to collect the impact-frequency tuples necessary to define the boundary between an unacceptable and acceptable business risk. For the sake of simplification and because the

information mentioned above is still unknown, the critical risk boundary was defined as the straight line (see Eq. 5.11) between the highest impact with the smallest frequency and the reciprocal point, namely the smallest impact with the highest frequency.

$$Im = -Im^{max} \cdot (1 - NPV) + Im^{max} \tag{5.11}$$

The highest impact, Im^{max}, was defined as the duration of a CIP/SIP batch resulting from the process recipe, which in this case study was 12 hours. The risk analysis was applied the prediction for both scenarios, as shown in **Table 5.2**. Values predicted failure times higher than the impact presented by Eq. (5.11) were rejected, values below the critical risk boundary were accepted. As a result, all the batches predicted as *about to fail* were accepted and the decision of interrupting the batch was enforced.

5.3.3 Maintenance of the classifier

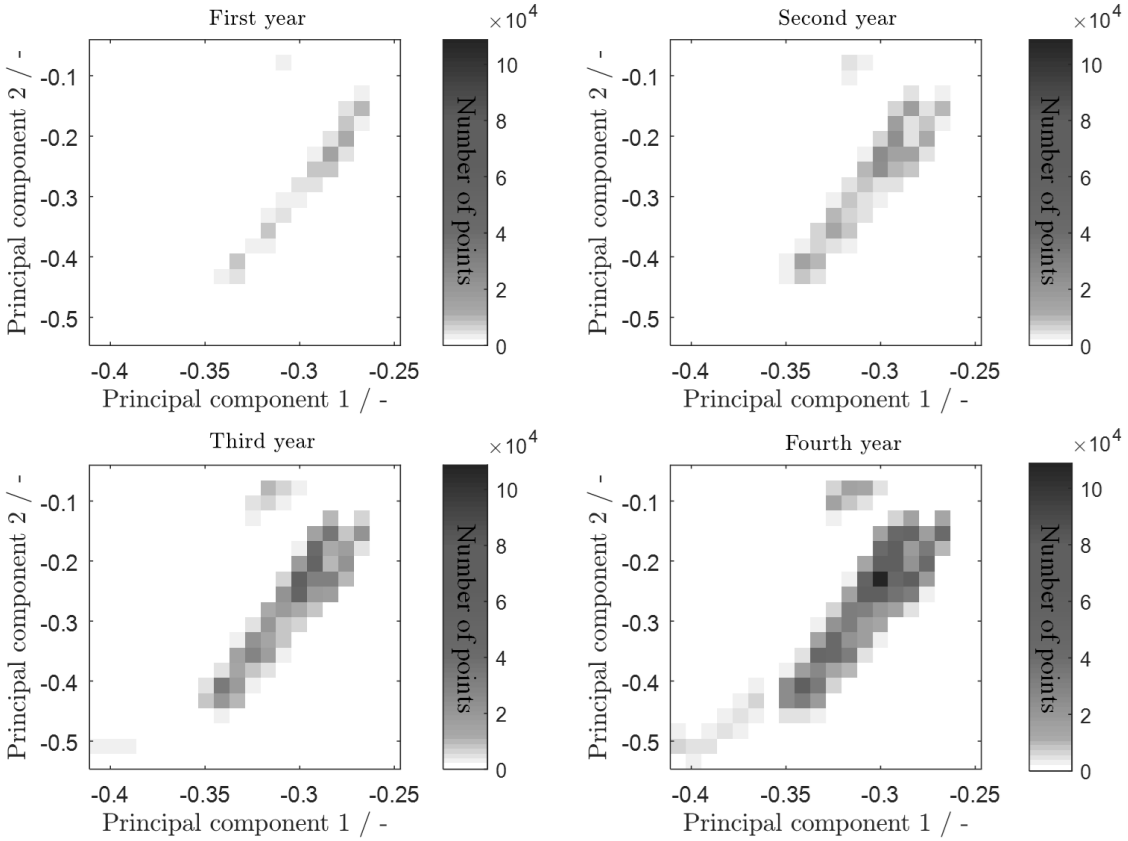


Figure 5.9 Evolution of the process space over the time.

As previously mentioned the process space evolved over time; **Figure 5.9** shows the evolution of process block *C*. The evolution of the process space of other blocks is shown in the appendix (see Appendix C.4). From **Figure**

5.9 it can be noticed that the process space is expanding over the time—i.e., white areas turn into blue—, which justified the employment of a time-based retraining policy.

The evolution of the model following the time-based maintenance policy was simulated for the available data (238 batches). A preliminary classifier at τ_0 was trained with the first three months of data, which contained the data of 15 batches; subsequently, the model was used to predict the presence of imminent failures and was retrained every $\Delta\tau$ equal to three months. The training dataset increases at each retraining activity; i.e., all new historical data were included in the updated training dataset every at interval $\Delta\tau$ in addition to the full set of previous training data. The simulation was performed for both the scenarios resulting in the salvaging of 12 batches over the period of 4 years. The complete results of the simulation are shown in **Table C.4.1** and **Table C.4.2** in the Appendix.

5.4 Result and discussion

The application and validity of the algorithm were shown in an implementation with real-world data recorded during a CIP/SIP process of a sterile filling facility. Two simulations for each risk scenario were performed in this study, namely the training and deployment of a DT classifier on the total training dataset (four years maturity) and the evolution of a classifier along a four years timespan. The results of the predictions were coupled with the decision-making tool providing the decision of interrupting the batch. The results of the simulation can be assessed in terms of the number of salvaged and lost batches, where the term lost refers to an interrupted batch that in reality would have been successful; a salvaged batch is a batch which failed in reality and was recognized as *about to fail* by the classifier. The time savings are also used to assess the impact of the algorithm on the operations; in case of salvaged batches, the time savings are defined as the difference between the time of detection and the real time of failure. In the case of a lost batch the time lost is equal to the time of detection of the failure.

First, the two deployed classifiers were evaluated by their performance and their capability of identifying imminent failure; the deployment resulted in the salvaging of three batches in both scenarios out of eight batches, and the loss of two batches in the risky scenario. The implementation of the algorithm could result in the

potential saving of 106 and 100 hours each month, which is approximately the timespan in which 16 batches are performed, for the conservative and the risky scenarios, respectively.

Second, the creation and evolution of two classifiers, conservative and risky, was simulated resulting in the salvage of 12 and 31 and loss of 12 and 34 batches over the total period of four years, respectively. The algorithm applied under these conditions over the four years of data did not reduce the actual downtime as in the previous example due to the high number of lost batches. This could be attributed to either the retraining interval being too large or the data still containing non-causally correlated data overlapping. From **Table C.4.2**, it can be noticed that the real-time decision-making part of the algorithm considered the unacceptable risk of taking a preventive action in the two cases, Without this tool the two Type II classification errors would have cost 20.41 h (10.77 h and 9.74 h, respectively; see **Table C.4.2**).

The study presented three main novelties, namely the supervised and conservative prediction of imminent failures, the intelligent retraining system, and the tool decision-making for considering business risk. First, a classification model was constructed for two scenarios, namely for conservative and risky decision-making. As expected from the sensitivity analysis, after the selection of the parameters γ and DOO_c the model did not deliver Type II classification errors after the training in the conservative scenario; on the contrary, in the risky scenario, Type II errors were observed. Second, it was observed that the machine-learning models required evolution—i.e., retraining—over time, which was shown by applying the time-based retraining part of the maintenance algorithm. Although it was not shown in this work, the *event-based retraining* and the *learning* parts of the maintenance algorithm were responsible for updating the models whenever new data points or labels were available. The calculation of the local DOO through grid search simplified the search of volumes neighboring the class boundaries and reduced the computational effort during the retraining activity. Third, the method for assessing the risk of intervening in a production batch as the risk of worsening the current production supported the decision of the policymakers, which were represented by the conservative and risky scenarios.
The algorithm presented three main limitations: First, the training of each model (two in total) required around 30 minutes on average, which become excessively long when the maintainance algorithm is implemented in commercial operations. The reason could be attributed to the time required to train the single block-specific sub-models (22 models in total) individually. The computational effort for training the model also increased with

increasing the size of the training dataset, which suggested a potential explosion of the computational time over the time. The limitation caused by the hardware could be solved by employing cluster computing, which was not accessible in this case because of data security constraints, in the training activity. Second, the data transformation part of the algorithm is simplistic; discretizing the process space during the relabeling step guarantees a specific decision-making strategy but decreases the sharpness of the decision boundary allowing more Type I classification errors. The algorithm could be improved for commercial applications by selecting alternative algorithms, such as nearest neighbor clustering, for selecting the local relabeling volumes. Third, the definition of the critical boundary is dependent on the decision maker. Thus the characterization of the business risk is subjective. Even though it is commonly applied, the assumption of a linear critical risk is simplistic does not reflect the reality of the business risk. To increase the contribution of the risk analysis to the final decision, a progressive critical risk with respect to the NPV could be applied. A progressive boundary of critical risk would allow higher detectability but block the predictions that carry too high impacts.

5.5 Conclusion

In this chapter, an intelligent algorithm for the real-time prediction of imminent failures for the cleaning and sterilization process in the biopharmaceutical manufacturing was presented. The algorithm, which is applied to real-world industrial data, consists of two main parts, a failure prediction and a real-time decision-making. The predictive model intelligently evolves with the always-changing process by updating itself whenever it fails, or new information is available, resulting in a powerful model-predictive monitoring tool. The decision-making tool is adaptable to the policy of the user, who can opt for a more conservative or a rather risky decision support. The application of the presented tool allows salvaging batches without changing the process or the process control system; hence, it avoids the time and monetary investments required to modify a process that is located in a GMP environment. The work presents a new take on the topic of process improvement and process control highlighting the utilization of the large amount of data without the need of the conventional process model.

Three specific novelties were presented in this work: first, the supervised and conservative imminent failure prediction model, which removes data ambiguity leveraging the local DOO provided to the classification model high reliability and applicability in a running commercial process. Second, the quantitative incorporation of business risk could support the real-time decision on taking actions during the execution of a commercial batch;

third, the implementation of an intelligent maintenance algorithm that analyzes the performance of the model and automatically decides a model retraining policy.

The algorithm was applied to an industrial case study in biopharmaceutical manufacturing with the goal of providing a tool for predicting imminent process failures. Two scenario analyses were performed by changing the risk of the decision; the scenarios represented a conservative decision-making and a risky decision-making policy. The two scenarios resulted in the potential salvage of 19% of the deployment batches; the potential time saving per month of 100 and 106 hours resulted for the first and second scenario, respectively. The time evolution and the temporal deployment of the model was simulated and resulted in the salvage of 12 batches in both scenarios; however, because of the high amount Type II error the effect on the downtime was minimal.

The algorithm still presents some limitations as mentioned above. Among others, the low specificity (type I classification error) is prominent and limits the applicability of the entire algorithm in real operations. The reason of the low specificity was attributed to presence of non-causal overlapping in the training data.

In future work it is of key importance to improve the prediction capability of the model; classification algorithm alternative to the decision tree model, such KNN or SVM, could improve the prediction performance and stability over time of the algorithm. Similarly, the further splitting of the data to a sub-block level, such as task, would remove non-causal data overlap improving the specificity of the prediction. Furthermore, more attention has to be invested in the definition of the critical business risk to better translate strategies and goals into quantitative indicators that can be used for supporting the real-time decision-making. Additionally, to further improve the applicability of the algorithm, future studies would have to invest efforts in improving the performance of the time-based retraining as well as include the retraining policies presented in the method section but not demonstrated in the case study. Finally, the descriptors and outcomes of the predictions, which were observed in the principal component dimension, have to be translated in the physical dimension. This would allow gaining additional process knowledge, and identifying and possibly removing the causes of process failures.

5.6 Nomenclature

A	Local area inside the process space	-

α_j	Age of the classifier at iteration j	h
B	Sudden shift parameter	-
C	Class variable, $C \in \{successful,\ about\ to\ fail\}$	
$C_{t_{n+1}}^{\text{real}}$	Real class at time t_{n+1}	
$C_{t_{n+1}}^{\text{pred}}$	Predicted class at time t_{n+1}	
D	Slope of the linear relation	Pa s^{-1}
d^+	Density of the points labeled as *about to fail*	-
DOO	Degree of overlap, $DOO \in [0,1]$	-
DOO_c	Critical degree of overlap	-
G	Time gain	h
$\hat{g}$	Estimated time gain for the local decision	h
γ	Granularity parameter	
Im	Impact of an intervention	h
Im^{max}	Maximum impact of an intervetion	h
I	Total number of sensors	-
i	Sensor counter $i \in \{1, I\}$	
j	Classifier training iteration counter	
M	Local total number of data points	-
n	Time counter	
NPV	Negative predictive value	-
P	Set of principal components $P \in \{p_1, p_2, \dots, p_I\}$	
p_i	Principal component i	-
t_n	General time point	h
$\hat{t}^{\text{failure}}$	Estimated time at which the failure will occur	h
τ_j	Training time point of the classifier	h
x_i	Value of the sensor i	K, Pa, …

x_i'	Scale value of the sensor i	-
$\tilde{x}_i$	Transformed value of the sensor i	-

Chapter 6: Conclusion

The thesis proposed a data-driven framework for supporting the decision-making in process and operation improvement for biopharmaceutical manufacturing. The framework consists of three main steps: the integration of existing industrial databases, assessment of the process performance and identification of the task to improve, and imminent failure mitigation by plant predictive maintenance. The second and third step of the framework tackle performance improvement through process re-design, which is long-term support, and through the real-time plant maintenance, which is intervention support, respectively. The third step connects the industrial data storage system to the tool developed in the other steps to ensure industrial compatibility of the framework; this step is essential for the bridging between industry and academia, which is one of the goals of the author. In comparison with other studies[134,164,172], this thesis provided a novel tool or approach that is applicable in an industrial environment and can solve long-term challenges as well as daily operation ones. This work integrated methods from various fields of research, such as data mining, machine learning, and natural language recognition. Moreover, the work could provide a multidisciplinary and multi-faceted view on the pharmaceutical world and the challenges found in this particular manufacturing sector. The data preprocessing algorithm utilized an extraneous perspective, namely the view of biotechnology, i.e., DNA sequencing, to solve a technical problem of computer science; the algorithm provided a multidisciplinary aspect to the thesis, which was one the author's statements stated in section 2.2. In the next sections, the conclusions of the three aspects mentioned in the introduction, namely, the impact on academia, industry, and society are presented singularly in more details.

6.1 Impact on the academia

As previously mentioned the framework and more specifically the data transformation algorithm showed the compatibility of the work to industrial operations and systems. This work introduced some general novelties in addition to the specific ones presented in **Chapters 3**, **4**, and **5**. First, uncertainty was defined from historical data and used to assess the process performance considering the presence of operators; contrary to the studies that only considered exogenous uncertainty in the evaluation of the performance,[130] the assessment considered the operators and human as part of the process and source of endogenous uncertainty, and provided a new perspective on process re-design. Second, in various occasions, e.g., noise classification, and the definition of the feasibility indicator, the framework could effectively integrate knowledge from experts in quantitative decision-making. Finally, the framework could also integrate the characters and the limiting factor that is very

peculiar to pharmaceutical manufacturing, namely GMP. GMP information was incorporated into the modeling of performance and in the decision-making mechanism, which is a value addition for such applied research.

6.2 Impact on the industry

The application of the framework in an industrial environment would bring several improvements in the decision-making process by simplifying and automating various activities. First, the data transformation algorithm, which is based on intelligent machine learning models, would automate tedious and obsolete tasks, such as data cleaning and structuring. Second, data historians would be optimally exploited; the data are used for creating models that support the process redesign and the monitoring and prediction of operations, and not used exclusively for retrospective analysis. Third, in addition to providing advanced mathematical methods, this work presented an integrated tool, which considers the risk of its implementation, for supporting the decision-making in manufacturing operation. Fourth, by introducing an advanced statistical method, event-specific process features were extrapolated and used for the reduction of unexpected downtime. The case study demonstrated that the framework is industrially compatible and that, if implemented in the operations, it could improve the decision-making process as well as the process performance.

Some limitations regarding the industrial applicability were identified in this work. Although the framework considers some simple aspects of GMP, a consideration of the entire GMP and its complex influence on process modification was not fully incorporated in the framework. Furthermore, given industrial implementation, the tool still requires extensive efforts in managing the integration of constraints from the informatics systems and the regulators. Finally, the extreme strong GMP culture, which is omnipresent in the pharmaceutical industry, and the lack of support from other studies, only allowed the employment of the framework to single or use cases or stand-alone process. A systemic or plant-wide implementation in the current decision-making process is still limited in the reality because of the reasons mentioned above.

Nonetheless, part of the results of the case study, namely, the what-if analysis of Scenario 2, was implemented in the industrial facility, leading to 120 h of time savings per year, which can be potentially utilized for manufacturing operations resulting in a 3% increase of the production capacity. Without considering the potential gain derived by the increase in capacity, the contribution of the study resulted in the reduction of the fixed costs of 0.5 million CHF per production line per year.

6.3 Impact on the society

This work is one of the many studies that aim to improve the global healthcare system by providing a solution to specific and technical problems. More specifically, in this study, the reduction of manufacturing costs by decreasing unexpected downtimes and increasing production capacity of high potent medications, such as anti-cancer drugs, was the focus.

Increasing production capacity, of course, has a high impact on the economy of a drug producer; in fact it guarantees that, the increasing number of new drugs entering the market can be manufactured without investing in new facilities. Ideally speaking, the cost of manufacturing can be reduced if the currently working facilities would be optimized instead of investing in new ones; therefore, the framework, especially the performance assessment step, is aimed at reducing manufacturing cost.

Reducing unexpected downtime, similar to the previous case, has repercussions on the profitability of a process; however, an essential aspect that has to be considered is the effect of unexpected downtime on the supply of a drug to the market. With more unexpected downtime, more reserve has to be maintained in storage to provide the necessary drug to the patient. The risk associated with keeping large inventory is the shelf life of the product, which decreases with time, increasing the risk of wasting life-saving products. The predictive maintenance part of the framework aims at reducing the unexpected downtime by predicting the process failures; by doing so, it indirectly decreases the risk of increasing waste.

The framework is relatively general and is adaptable to several processes; however, because the production of pharmaceutical products is a very complicated process, only the influence of the single improvement on the society could be deduced.

To show the effect of the framework from the two perspectives, industrial case studies were performed: The first resulted in the identification of unnecessary process steps, and the second showed that the algorithm could save up to 100 hours production time per month if applied in real-time decision making. It can be concluded that the implementation of the framework on the most significant scale would result in the potential reduction of manufacturing cost and the increase of drug availability on the market. Not to be forgotten is that the improvement of a process in general usually results in improvement of their efficiency, which also has a positive effect from an environmental stand point.

Chapter 7: Outlook

In future works, the framework has to be systematically implemented in the manufacturing, after the necessary adaptations, such as the creation of a user interface, and further generalization to increase flexibility. Moreover, a long-term benefit analysis has to be performed to prove the outcome of the decisions over time; lastly, the framework has to be expanded to other processes that are not CIP/SIP, such as filling, and transportation.

Another perspective, which has not yet been considered, is the contribution that such a framework can deliver in improving the sustainability of the manufacturing process. By redefining the KPI through sustainability indicator, it is possible to analyze the impact of process improvement and operations support on the environment. The addition of such an assessment will enable the framework to deliver multi-objective decision-making, considering the economic, supply risk-related, political, and environmental perspectives.

7.1 Relevance of the study to the outside of pharmaceutical manufacturing

The framework presented in this work is very general and can be adapted to the need of each user independently on the industry sector. In future works, the framework has to be applied to other processes, such as the downstream process of DS or the sterile filling process, and in other industry sectors, such as watch, car, and discrete manufacturing. The author is convinced that the framework can be transformed into every process consisting of a series of tasks through the reinterpretation of industry-related indicators such as the feasibility indicator. Additionally, the knowledge collected using the framework in assessing existing facilities, especially the knowledge about uncertainty, will support an advanced grass root design of new process. Such as an evolved root design will consider enlarged design boundary compared with the conventional design; in fact, additionally to the conventional boundaries, it will consider the operational environment in which the new process is located. Furthermore, the implementation of the algorithm presented in **Chapter 5** could result in a very hands-on and profitable tool, which will reduce the downtime as well as guarantee the supply reliability required in the market.

7.2 Industry 4.0 and Internet of Things concepts

As mentioned in the conclusion sections in **Chapters 3**, **4**, and **5**, the study provided an introductory lesson to I4.0- and IoT tools, which until now, were unknown to the pharmaceutical industry. Following the current trend on data production, the data inventory, comprising manufacturing companies, is increasing exponentially; hence, such a framework can set the practical basis for the development of further integrated tools. The

expansion from a process-wide to plant-wide predictive monitoring and process improvement, by installing the new sensor and optimizing the position of old ones could increase the value of CAPE/PSE methods in the industry.

Another technology, which is still in its infancy but is suitable for pharmaceutical operations support, is blockchain technology; such technology could further facilitate the introduction of the framework by dealing with GMP controlled activity and information. The author is convinced that the road toward smart pharmaceutical manufacturing is still long, but at least we are on the right track and already progressing.

Literature cited

Literature cited

1. Eurostat. Mortality and life expectancy statistics. *Mortal Life Expect Stat*. 2016:2015-2018.
2. Eurostat. Fertility statistics.http://ec.europa.eu/eurostat/statistics-explained/index.php/Fertility_statistics. Published 2012. Accessed April 6, 2018
3. Eurostat. Statistics explained. Population structure and ageing.http://ec.europa.eu/eurostat/statistics-explained/index.php/Population_structure_and_ageing. Published 2017. Accessed April 6, 2018
4. Goldstone JA. Flash points and tipping points: Security implications of global population. *Popul Chang Glob Secur*. 2006.
5. Boecking W, Klamar A, Kitzmann F, Kirch W. Pharmaco-economic impact of demographic change on pharmaceutical expenses in Germany and France. *BMC Public Health*. 2012;12:894.
6. IFPMA. *The Pharmaceutical Industry and Global Health Facts and Figures 2017*.; 2017.
7. EFPIA. *The Pharmaceutical Industry in Figures: Key Data 2017*.; 2017.
8. Basu P, Joglekar G, Rai S, Suresh P, Vernon J. Analysis of manufacturing costs in pharmaceutical companies. *J Pharm Innov*. 2008;3:30-40.
9. Evaluate Ltd. World preview 2017, outlook to 2022. *Eval Pharma*. 2017:1-41.
10. Cabinet office - Council of economic and fiscal policy. Basic policy on economic and fiscal management and reform 2017. 2017.
11. Godman B, Wettermark B, Bishop I, et al. European payer initiatives to reduce prescribing costs through use of generics. *Generics Biosimilars Initiat J*. 2012;1:22-27.
12. Vogler S. The impact of pharmaceutical pricing and reimbursement policies on generics uptake: implementation of policy options on generics in 29 European countries─an overview. *Generics Biosimilars Initiat J*. 2012;1:93-100.
13. Lipsitz YY, Timmins NE, Zandstra PW. Quality cell therapy manufacturing by design. *Nat Biotechnol*. 2016;34:393-400.
14. Vogenberg FR, Isaacson Barash C, Pursel M. Personalized medicine: part 1: evolution and development into theranostics. *P T*. 2010;35:560-576.
15. Sugiyama H, Schmidt R. Realizing continuous improvement in pharmaceutical technical operations -

Business process model in Roche's parenterals production Kaiseraugst. *Comput Aided Chem Eng*. 2012;30:422-426.

16. Jozala AF, Geraldes DC, Tundisi LL, et al. Biopharmaceuticals from microorganisms: from production to purification. *Brazilian J Microbiol*. 2016;47:51-63.
17. Meyer BK, Coless L. Compounding and filling: Drug substance to drug product. In Meyer BK, Therapeutic protein drug products: Practical approaches to formulation in the laboratory, manufacturing, and the clinic. Sawston, Cambridge: Woodhead Publishing Limited; 2012:83-95.
18. Unger-Bimczok B, Kosian T, Kottke V, Hertel C, Rauschnabel J. Hydrogen peroxide vapor penetration into small cavities during low-temperature decontamination cycles. *J Pharm Innov*. 2011;6:32-46.
19. Seyfarth H. The new FDA aseptic guidance - Part 3: Qualification/Validation. *Pharm Ind*. 2005;67:1488-1501.
20. U.S. Food & Drug Administration. CFR - Code of federal regulations title 21.https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?fr=211.25. Published 2016. Accessed April 9, 2018
21. U.S. Food & Drug Administration. CFR - Code of federal regulations title 21.https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?fr=211.113. Published 2016. Accessed April 10, 2018
22. U.S. Food & Drug Administration. CFR - Code of federal regulations title 21.https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?fr=211.188. Published 2016. Accessed April 10, 2018
23. U.S. Food & Drug Administration. CFR - Code of federal regulations title 21.https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?fr=211.110. Published 2016. Accessed April 10, 2018
24. U.S. Food & Drug Administration. CFR - Code of federal regulations title 21.https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?fr=211.165. Published 2016. Accessed April 10, 2018
25. U.S. Food & Drug Administration. CFR - Code of federal regulations title

21.https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?fr=211.192. Published 2016. Accessed April 10, 2018

26. Sugiyama H, Schmidt R. Business model of continuous improvement in pharmaceutical production processes. In Kraslawski A, Turunen I, *Computer Aided Chemical Engineering*. Vol 32. Elsevier; 2013:697-702.

27. Sesen MB, Suresh P, Banares-Alcantara R, Venkatasubramanian V. An ontological framework for automated regulatory compliance in pharmaceutical manufacturing. *Comput Chem Eng*. 2010;34:1155-1169.

28. Boltic Z, Jovanovic M, Petrovic S, Bozanic V, Mihajlovic M. Continuous improvement concepts as a link between quality assurance and implementation of cleaner production: Case study in the generic pharmaceutical industry. *Chem Ind Chem Eng Q*. 2016;22:55-64.

29. Casola G, Sugiyama H, Siegmund C, Mattern M. Uncertainty-conscious methodology for process performance assessment in biopharmaceutical drug product manufacturing. *AIChE J*. 2018;64:1272-1284.

30. Bonfill A, Espuña A, Puigjaner L. Decision support framework for coordinated production and transport scheduling in SCM. *Comput Chem Eng*. 2008;32:1214-1232.

31. Laínez JM, Reklaitis G V., Puigjaner L. Linking marketing and supply chain models for improved business strategic decision support. *Comput Chem Eng*. 2010;34:2107-2117.

32. Suresh P, Hsu S-H, Akkisetty P, Reklaitis G V., Venkatasubramanian V. OntoMODEL: Ontological mathematical modeling knowledge management in pharmaceutical product development, 1: Conceptual framework. *Ind Eng Chem Res*. 2010;49:7758-7767.

33. Li Z, Ierapetritou M. Process scheduling under uncertainty: Review and challenges. *Comput Chem Eng*. 2008;32:715-727.

34. Yunus NA, Gernaey K V., Woodley J, Gani R. A systematic methodology for design of tailor-made blended products. *Comput Chem Eng*. 2014;66:201-213.

35. Laínez JM, Hegyháti M, Friedler F, Puigjaner L. Using S-graph to address uncertainty in batch plants. *Clean Technol Environ Policy*. 2010;12:105-115.

36. Pirola C, Galimberti M, Comazzi A, Bozzano G, Hillestad M, Manenti F. Integrated reactor staging and plant optimization of a Biomass-To-Liquid technology. *Comput Chem Eng*. 2017;106:719-729.

37. Martín M, Grossmann IE. Towards zero CO2 emissions in the production of methanol from switchgrass. CO2 to methanol. *Comput Chem Eng*. 2017;105:308-316.

38. Li M. Systematic analysis and optimization of power generation in pressure retarded osmosis: effect of multistage design. 2017.

39. Elsholkami M, Elkamel A, Vargas F. Optimized integration of renewable energy technologies into Alberta's oil sands industry. *Comput Chem Eng*. 2016;90:1-22.

40. Venkatasubramanian V, Rengaswamy R, Yin K, Kavuri SN. A review of process fault detection and diagnosis: Part I: Quantitative model-based methods. *Comput Chem Eng*. 2003;27:293-311.

41. Biegler LT, Grossmann IE. Retrospective on optimization. *Comput Chem Eng*. 2004;28:1169-1192.

42. Boukouvala F, Muzzio FJ, Ierapetritou MG. Dynamic data-driven modeling of pharmaceutical processes. *Ind Eng Chem Res*. 2011;50:6743-6754.

43. Biscarri F, Monedero I, León C, Guerrero JI, González R, Pérez-Lombard L. A decision support system for consumption optimization in a naphtha reforming plant. *Comput Chem Eng*. 2012;44:1-10.

44. Muñoz E, Capón-garcía E, Hungerbühler K, Espuña A, Puigjaner L. Decision making support based on a process engineering ontology for waste treatment plant optimization. 2013;11:261-270.

45. Ning C, You F. Data-driven stochastic robust optimization: General computational framework and algorithm leveraging machine learning for optimization under uncertainty in the big data era. *Comput Chem Eng*. 2018;111:115-133.

46. Kadlec P, Gabrys B, Strandt S. Data-driven soft sensors in the process industry. *Comput Chem Eng*. 2009;33:795-814.

47. Kadlec P, Grbić R, Gabrys B. Review of adaptation mechanisms for data-driven soft sensors. *Comput Chem Eng*. 2011;35:1-24.

48. Thomson Reuters. Web of Science [v.5.28] - Web of Science core collection result analysis.http://wcs.webofknowledge.com/RA/analyze.do. Published 2018. Accessed April 10, 2018

49. Reklaitis G. Perspectives on process systems engineering R&D in support of pharmaceutical product /

process development and manufacturing. In *Computer Aided Chemical Engineering*. Vol 24. ; 2007:1-4.

50. Gani R. Chemical product design: Challenges and opportunities. *Comput Chem Eng*. 2004;28:2441-2457.
51. Gebreslassie BH, Diwekar UM. Homogenous multi-agent optimization for process systems engineering problems with a case study of computer aided molecular design. *Chem Eng Sci*. 2017;159:194-206.
52. Jiménez-González C, Woodley JM. Bioprocesses: Modeling needs for process evaluation and sustainability assessment. *Comput Chem Eng*. 2010;34:1009-1017.
53. Wang Z, Escotet-Espinoza MS, Ierapetritou M. Process analysis and optimization of continuous pharmaceutical manufacturing using flowsheet models. *Comput Chem Eng*. 2017;107:77-91.
54. Vieira M, Pinto-Varela T, Moniz S, Barbosa-Póvoa AP, Papageorgiou LG. Optimal planning and campaign scheduling of biopharmaceutical processes using a continuous-time formulation. *Comput Chem Eng*. 2016;91:422-444.
55. Singh R, Sahay A, Muzzio F, Ierapetritou M, Ramachandran R. A systematic framework for onsite design and implementation of a control system in a continuous tablet manufacturing process. *Comput Chem Eng*. 2014;66:186-200.
56. Rantanen J, Khinast J. The future of pharmaceutical manufacturing sciences. *J Pharm Sci*. 2015;104:3612-3638.
57. Sajjia M, Shirazian S, Egan D, et al. Mechanistic modelling of industrial-scale roller compactor 'Freund TF-MINI model.' *Comput Chem Eng*. 2017;104:141-150.
58. Casola G, Yoshikawa S, Nakanishi H, Hirao M, Sugiyama H. Systematic retrofitting methodology for pharmaceutical drug purification processes. *Comput Chem Eng*. 2015;80.
59. Boukouvala F, Niotis V, Ramachandran R, Muzzio FJ, Ierapetritou M. An integrated approach for dynamic flowsheet modeling and sensitivity analysis of a continuous tablet manufacturing process. *Comput Chem Eng*. 2012;42:30-47.
60. Singh R, Ierapetritou M, Ramachandran R. System-wide hybrid MPC-PID control of a continuous pharmaceutical tablet manufacturing process via direct compaction. *Eur J Pharm Biopharm*.

2013;85:1164-1182.

61. Köksal G, Batmaz I, Testik MC. A review of data mining applications for quality improvement in manufacturing industry. *Expert Syst Appl*. 2011;38:13448-13467.

62. Papadakis E, Anantpinijwatna A, Woodley J, Gani R. A reaction database for small molecule pharmaceutical processes integrated with process information. *Processes*. 2017;5:58.

63. Diab S, Gerogiorgis DI. Process modelling, simulation and technoeconomic evaluation of crystallisation antisolvents for the continuous pharmaceutical manufacturing of rufinamide. *Comput Chem Eng*. 2018;111:102-114.

64. Rasche ML, Jiang M, Braatz RD. Mathematical modeling and optimal design of multi-stage slug-flow crystallization. *Comput Chem Eng*. 2016;95:240-248.

65. Ridder BJ, Majumder A, Nagy ZK. Parametric, optimization-based study on the feasibility of a multisegment antisolvent crystallizer for in situ fines removal and matching of target size distribution. *Ind Eng Chem Res*. 2016;55:2371-2380.

66. Luo H, Yang R, Zhao Y, et al. Recent advances and strategies in process and strain engineering for the production of butyric acid by microbial fermentation. *Bioresour Technol*. 2018;253:343-354.

67. Liu S, Farid SS, Papageorgiou LG. Integrated optimization of upstream and downstream processing in biopharmaceutical manufacturing under uncertainty: A chance constrained programming approach. *Ind Eng Chem Res*. 2016;55:4599-4612.

68. Martinetz MC, Rehrl J, Aigner I, Sacher S, Khinast J. A continuous operation concept for a rotary tablet press using mass flow operating points. *Chemie Ing Tech*. 2017;89:1006-1016.

69. Sajjia M, Shirazian S, Kelly CB, Albadarin AB, Walker G. ANN analysis of a roller compaction process in the pharmaceutical industry. *Chem Eng Technol*. 2017;40:487-492.

70. Içten E, Joglekar G, Wallace C, et al. Knowledge provenance management system for a dropwise additive manufacturing system for pharmaceutical products. *Ind Eng Chem Res*. 2016;55:9676-9686.

71. Su Q, Moreno M, Giridhar A, Reklaitis G V., Nagy ZK. A systematic framework for process control design and risk analysis in continuous pharmaceutical solid-dosage manufacturing. *J Pharm Innov*. 2017;12:327-346.

72. Bosca S, Fissore D, Demichela M. Risk-based design of a freeze-drying cycle for pharmaceuticals. *Ind Eng Chem Res*. 2015;54:12928-12936.

73. Eberle L, Sugiyama H, Papadokonstantakis S, Graser A, Schmidt R, Hungerbühler K. Data-driven tiered procedure for enhancing yield in drug product manufacturing. *Comput Chem Eng*. 2016;87:82-94.

74. Rogers A, Ierapetritou M. Challenges and opportunities in pharmaceutical manufacturing modeling and optimization. In Mario E, John S, Gavin T, *Computer Aided Chemical Engineering*. Vol 34. Elsevier; 2014:144-149.

75. Gartner Inc. Gartner IT glossary.https://www.gartner.com/it-glossary/digitalization/. Published 2018. Accessed April 11, 2018

76. Brennen S, Kreiss D. Digitalization and Digitization. *Cult Digit*. 2014:1-13.

77. Ross J MIT S. Don't confuse Digital with Digitization. *MIT Sloan*. 2017:1-5.

78. Tresp V, Marc Overhage J, Bundschus M, Rabizadeh S, Fasching PA, Yu S. Going digital: A survey on digitalization and large-scale data analytics in healthcare. *Proc IEEE*. 2016;104:2180-2206.

79. Maedche A. Interview with Wolfgang Gaertner on "Digitalization in retail banking: Differentiation and standardization through IT." *Bus Inf Syst Eng*. 2015;57:83-85.

80. Sia KS, Soh C, Weill P. How DBS bank pursued a digital business strategy. *MIS Q Exec*. 2016;15:105-121.

81. Peters S, Chun J-H, Lanza G. Digitalization of automotive industry – scenarios for future manufacturing. *Manuf Rev*. 2016;3:1.

82. Tao F, Zhang L, Nee AYC. A review of the application of grid technology in manufacturing. *Int J Prod Res*. 2011;49:4119-4155.

83. Blaya F, Pedro PS, Silva JL, D'Amato R, Heras ES, Juanes JA. Design of an orthopedic product by using additive manufacturing technology: The arm splint. *J Med Syst*. 2018;42:54.

84. Reitze A, Jürgensmeyer N, Lier S, Kohnke M, Riese J, Grünewald M. Roadmap for a Smart Factory: A Modular, Intelligent Concept for the Production of Specialty Chemicals. *Angew Chemie Int Ed*. 2018;57:4242-4247.

85. Lepage D, Jiménez A, Beauvais J, Dubowski JJ. Real-time detection of influenza A virus using

semiconductor nanophotonics. *Light Sci Appl*. 2013;2:e62-e62.

86. Schenkendorf R. Supporting the shift towards continuous pharmaceutical manufacturing by condition monitoring. In *2016 3rd Conference on Control and Fault-Tolerant Systems (SysTol)*. IEEE; 2016:593-598.
87. Li D. Perspective for smart factory in petrochemical industry. *Comput Chem Eng*. 2016;91:136-148.
88. Clemons J. Creating value from smart manufacturing.http://www.pharmamanufacturing.com/articles/2016/creating-value-from-smart-manufacturing/?show=all. Published 2016. Accessed August 3, 2017
89. Kemppainen P. Pharma digitalisation: challenges and opportunities in transforming the pharma industry.https://www.europeanpharmaceuticalreview.com/news/51733/pharma-digitalisation-challenges/. Published 2017. Accessed August 3, 2017
90. FDA. Current good manufacturing practice for finished pharmaceuticals.https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=211. Published 2016. Accessed June 26, 2017
91. FDA. *Data Integrity and Compliance with CGMP Guidance for Industry*. Silver Spring; 2016.
92. Zhao D, Li S. A 3D image processing method for manufacturing process automation. *Comput Ind*. 2005;56:975-985.
93. Martín Ó, Ahedo V, Santos JI, De Tiedra P, Galán JM. Quality assessment of resistance spot welding joints of AISI 304 stainless steel based on elastic nets. *Mater Sci Eng A*. 2016;676:173-181.
94. Pereda M, Santos JI, Martín Ó, Galán JM. Direct quality prediction in resistance spot welding process: Sensitivity, specificity and predictive accuracy comparative analysis. *Sci Technol Weld Join*. 2015;20:679-685.
95. Kuo C-FJ, Juang Y. A study on the recognition and classification of embroidered textile defects in manufacturing. *Text Res J*. 2016;86:393-408.
96. Kusiak A. Data mining: manufacturing and service applications. *Int J Prod Res*. 2006;44:4175-4191.
97. Ma CY, Wang XZ. Inductive data mining based on genetic programming: Automatic generation of decision trees from data for process historical data analysis. *Comput Chem Eng*. 2009;33:1602-1616.

98. Simon LL, Hungerbühler K. Industrial batch dryer data mining using intelligent pattern classifiers: Neural network, neuro-fuzzy and Takagi-Sugeno fuzzy models. *Chem Eng J*. 2010;157:568-578.

99. Myers EW. A sublinear algorithm for approximate keyword searching. *Algorithmica*. 1994;12:345-374.

100. Lewinski NA, McInnes BT. Using natural language processing techniques to inform research on nanotechnology. *Beilstein J Nanotechnol*. 2015;6:1439-1449.

101. Ukkonen E. Algorithms for approximate string matching. *Inf Control*. 1985;64:100-118.

102. Wagner RA, Fischer MJ. The string-to-string correction problem. *J ACM*. 1974;21:168-173.

103. Bunke H, Csirik J. Parametric string edit distance and its application to pattern recognition. *IEEE Trans Syst Man Cybern*. 1995;25:202-206.

104. Eberle LG, Sugiyama H, Schmidt R. Improving lead time of pharmaceutical production processes using Monte Carlo simulation. *Comput Chem Eng*. 2014;68:255-263.

105. Meneghetti N, Facco P, Bezzo F, Himawan C, Zomer S, Barolo M. Knowledge management in secondary pharmaceutical manufacturing by mining of data historians-"A proof-of-concept study. *Int J Pharm*. 2016;505:394-408.

106. Singh J, Gupta V. A systematic review of text stemming techniques. *Artif Intell Rev*. 2017;48:157-217.

107. Koonin EV, Novozhilov AS. Origin and evolution of the genetic code: the universal enigma. *IUBMB Life*. 2009;61:99-111.

108. Fernández-Delgado M, Cernadas E, Barro S, Amorim D, Amorim Fernández-Delgado D. Do we need hundreds of classifiers to solve real world classification problems? *J Mach Learn Res*. 2014;15:3133-3181.

109. Powers D. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness & correlation. *J Mach Learn Technol*. 2011;2:37-63.

110. Macqueen J. Some methods for classification and analysis of multivariate observations. *Proc Fifth Berkeley Symp Math Stat Probab*. 1967;1:281-297.

111. Masek WJ, Paterson MS. A faster algorithm computing string edit distances. *J Comput Syst Sci*. 1980;20:18-31.

112. Boltic Z, Jovanovic M, Petrovic S, Bozanic V, Mihajlovic M. Continuous improvement concepts as a

link between quality assurance and implementation of cleaner production: Case study in the generic pharmaceutical industry. *Chem Ind Chem Eng Q*. 2016;22:55-64.

113. Simon K. The aause and effect (a.k.a. fishbone) diagram.https://www.isixsigma.com/tools-templates/cause-effect/cause-and-effect-aka-fishbone-diagram/. Published 2000. Accessed October 18, 2017

114. Alsyouf I, Al-Aomar R, Al-Hamed H, Qiu X. A framework for assessing the cost effectiveness of lean tools. *Eur J Ind Eng*. 2011;5:170-197.

115. Dünnebier G. Troubleshooting and process optimisation by integrating CAPE tools and Six Sigma methodology. In Braunschweig B, Joulia X, *Computer Aided Chemical Engineering*. Vol 25. Elsevier; 2008:943-948.

116. Jones EC, Parast MM, Adams SG. A framework for effective Six Sigma implementation. *Total Qual Manag Bus Excell*. 2010;21:415-424.

117. Dassau E, Zadok I, Lewin DR. Combining six-sigma with integrated design and control for yield enhancement in bioprocessing. *Ind Eng Chem Res*. 2006;45:8299-8309.

118. Casola G, Eberle LG, Siegmund C, Mattern M, Sugiyama H. Mid-term scheduling model based on state-task-network for considering plant specific constraints in pharmaceutical manufacturing. In Kravanja Z, *Computer Aided Chemical Engineering*. Vol 38. Elsevier; 2016:1141-1146.

119. Ciavotta M, Meloni C, Pranzo M. Scheduling dispensing and counting in secondary pharmaceutical manufacturing. *AIChE J*. 2009;55:1161-1170.

120. Reklaitis G, Khinast J, Muzzio FJ. Pharmaceutical engineering science—New approaches to pharmaceutical development and manufacturing. *Chem Eng Sci*. 2010;65:iv-vii.

121. Sundaramoorthy A, Li X, Evans JMB, Barton PI. Capacity planning for continuous pharmaceutical manufacturing facilities. In Karimi I, Srinivasan R, *Computer Aided Chemical Engineering*. Vol 31. Elsevier; 2012:1135-1139.

122. Costa A. Hybrid genetic optimization for solving the batch-scheduling problem in a pharmaceutical industry. *Comput Ind Eng*. 2015;79:130-147.

123. Jolliffe HG, Gerogiorgis DI. Plantwide design and economic evaluation of two continuous

pharmaceutical manufacturing (CPM) cases: Ibuprofen and artemisinin. *Comput Chem Eng*. 2016;91:260-288.

124. Jolliffe HG, Gerogiorgis DI. Process modelling and simulation for continuous pharmaceutical manufacturing of ibuprofen. *Chem Eng Res Des*. 2015;97:175-191.

125. Onyango V. Enhancing environmental integration in strategic environmental assessment (SEA): insight from sensitivity analysis. *J Environ Plan Manag*. 2015;59:1-19.

126. Bahakim SS, Rasoulian S, Ricardez-Sandoval LA. Optimal design of large-scale chemical processes under uncertainty: A ranking-based approach. *AIChE J*. 2014;60:3243-3257.

127. Cadini F, Gioletta A. A Bayesian Monte Carlo-based algorithm for the estimation of small failure probabilities of systems affected by uncertainties. *Reliab Eng Syst Saf*. 2016;153:15-27.

128. Grossmann I. Enterprise-wide optimization: A new frontier in process systems engineering. *AIChE J*. 2005;51:1846-1857.

129. Grossmann IE, Apap RM, Calfa BA, García-Herreros P, Zhang Q. Recent advances in mathematical programming techniques for the optimization of process systems under uncertainty. *Comput Chem Eng*. 2016;91:3-14.

130. Applequist GE, Pekny JF, Reklaitis G. Risk and uncertainty in managing chemical manufacturing supply chains. *Comput Chem Eng*. 2000:2211-2222.

131. Chhatre S, Francis R, Newcombe AR, et al. Global sensitivity analysis for the determination of parameter importance in the chromatographic purification of polyclonal antibodies. *J Chem Technol Biotechnol*. 2008;83:201-208.

132. Lakhdar K, Papageorgiou LG. An iterative mixed integer optimisation approach for medium term planning of biopharmaceutical manufacture under uncertainty. *Chem Eng Res Des*. 2008;86:259-267.

133. Lakerveld R, Benyahia B, Braatz RD, Barton PI. Model-based design of a plant-wide control strategy for a continuous pharmaceutical plant. *AIChE J*. 2013;59:3671-3685.

134. Li YF, Venkatasubramanian V. Leveraging bayesian approach to predict drug manufacturing performance. *J Pharm Innov*. 2016;11:1-8.

135. García-Muñoz S, Mercado J. Optimal selection of raw materials for pharmaceutical drug product design

and manufacture using mixed integer nonlinear programming and multivariate latent variable regression models. *Ind Eng Chem Res*. 2013;52:5934-5942.

136. Casola G, Yoshikawa S, Nakanishi H, Hirao M, Sugiyama H. Systematic retrofitting methodology for pharmaceutical drug purification processes. *Comput Chem Eng*. 2015;80:177-188.
137. Ross TD. Applications and extensions of SADT. *Computer (Long Beach Calif)*. 1985;18:25-34.
138. Casola G, Siegmund C, Mattern M, Sugiyama H. Integrated process performance assessment considering uncertainty in biopharmaceutical manufacturing operations. In Espuña A, *Computer Aided Chemical Engineering*. Elsevier; 2017:accepted for pubblication.
139. Savolainen J. Global sensitivity analysis of a feedback-controlled stochastic process model. *Simul Model Pract Theory*. 2013;36:1-10.
140. Nadaraya EA. On estimating regression. *Theory Probab Its Appl*. 1964;9:157-159.
141. Kanji GK. *100 Statistical Tests*. London: SAGE Publications Ltd; 2006.
142. Arwade SR, Moradi M, Louhghalam A. Variance decomposition and global sensitivity for structural systems. *Eng Struct*. 2010:1-10.
143. Helton JC, Davis FJ. Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. *Reliab Eng Syst Saf*. 2003;81:23-69.
144. Schmidt-Bader T. PAT und QbD im regulatorischen Umfeld der Pharmazeutischen Industrie. *Chemie Ing Tech*. 2010;82:415-428.
145. Hinz DC. Process analytical technologies in the pharmaceutical industry: the FDA's PAT initiative. *Anal Bioanal Chem*. 2006;384:1036-1042.
146. Rajalahti T, Kvalheim OM. Multivariate data analysis in pharmaceutics: A tutorial review. *Int J Pharm*. 2011;417:280-290.
147. De Beer TRM, Vercruysse P, Burggraeve A, et al. In-line and real-time process monitoring of a freeze drying process using Raman and NIR spectroscopy as complementary process analytical technology (PAT) tools. *J Pharm Sci*. 2009;98:3430-3446.
148. Wu H, Khan MA. Quality-by-Design (QbD): An integrated Process Analytical Technology (PAT) approach for real-time monitoring and mapping the state of a pharmaceutical coprecipitation process. *J*

Pharm Sci. 2010;99:1516-1534.

149. Kimura S-I, Iwao Y, Ishida M, et al. Evaluation of the physicochemical properties of fine globular granules prepared by a multi-functional rotor processor. *Chem Pharm Bull*. 2014;62:309-315.

150. O'Mahony N, Murphy T, Panduru K, Riordan D, Walsh J. Machine learning algorithms for process analytical technology. In *2016 World Congress on Industrial Control Systems Security (WCICSS)*. IEEE; 2016:1-7.

151. Burbidge R, Trotter M, Buxton B, Holden S. Drug design by machine learning: support vector machines for pharmaceutical data analysis. *Comput Chem*. 2001;26:5-14.

152. Hou T, Wang J, Zhang W, Wang W, Xu X. Recent Advances in Computational Prediction of Drug Absorption and Permeability in Drug Discovery. *Curr Med Chem*. 2006;13:2653-2667.

153. Gawehn E, Hiss JA, Schneider G. Deep Learning in Drug Discovery. *Mol Inform*. 2016;35:3-14.

154. Ding H, Mamitsuka H, Zhu S. Similarity-based machine learning methods for predicting drug^target interactions: a brief review.

155. Ekins S. The Next Era: Deep Learning in Pharmaceutical Research. *Pharm Res*. 2016;33:2594-2603.

156. Gams M, Horvat M, Ožek M, Luštrek M, Gradišek A. Integrating Artificial and Human Intelligence into Tablet Production Process. *AAPS PharmSciTech*. 2014;15:1447-1453.

157. Akseli I, Xie J, Schultz L, et al. A Practical Framework Toward Prediction of Breaking Force and Disintegration of Tablet Formulations Using Machine Learning Tools. *J Pharm Sci*. 2017;106:234-247.

158. Caggiano A, Segreto T, Teti R. Cloud manufacturing framework for smart monitoring of machining. In *Procedia CIRP*. Vol 55. Elsevier; 2016:248-253.

159. Wu D, Jennings C, Terpenny J, Kumara S. Cloud-based machine learning for predictive analytics: Tool wear prediction in milling. In *Proceedings - 2016 IEEE International Conference on Big Data, Big Data 2016*. IEEE; 2016:2062-2069.

160. Lei X, Sandborn PA. Maintenance scheduling based on remaining useful life predictions for wind farms managed using power purchase agreements. *Renew Energy*. 2018;116:188-198.

161. Verbert K, De Schutter B, Babuška R. Timely condition-based maintenance planning for multi-component systems. *Reliab Eng Syst Saf*. 2017;159:310-321.

162. Gao R, Wang L, Teti R, et al. Cloud-Enabled Prognosis for Manufacturing. *CIRP Ann - Manuf Technol*. 2015.

163. Susto GA, Schirru A, Pampuri S, McLoone S, Beghi A. Machine learning for predictive maintenance: A multiple classifier approach. *IEEE Trans Ind Informatics*. 2015;11:812-820.

164. Roy K, Undey C, Mistretta T, Naugle G, Sodhi M. Multivariate statistical monitoring as applied to clean-in-place (CIP) and steam-in-place (SIP) operations in biopharmaceutical manufacturing. *Biotechnol Prog*. 2014;30:505-515.

165. Xia T, Jin X, Xi L, Zhang Y, Ni J. Operating load based real-time rolling grey forecasting for machine health prognosis in dynamic maintenance schedule. *J Intell Manuf*. 2015;26:269-280.

166. Hohenegger J. Weighted standardization—A general data transformation method proceeding classification procedures. *Biometrical J*. 1986;28:295-303.

167. Mehrabi S, Mohammadi I, Kunjan K, Kharrazi H. Effects of data transformation methods on classification of patients diagnosed with myocardial infarction. *Stud Health Technol Inform*. 2013;192:1203.

168. Maadooliat M, Huang JZ, Hu J. Integrating data transformation in principal components analysis. *J Comput Graph Stat*. 2015;24:84-103.

169. Hang Cao X, Stojkovic I, Obradovic Z. A robust data scaling algorithm to improve classification accuracies in biomedical data. *BMC Bioinformatics*. 2016;17.

170. Hanif A, Azhar N. Resolving class imbalance and feature selection in customer churn dataset. In *2017 International Conference on Frontiers of Information Technology (FIT)*. IEEE; 2017:82-86.

171. Wenyin Tang, Mao KZ, Lee Onn Mak, Gee Wah Ng. Classification for overlapping classes using optimized overlapping region detection and soft decision. In *2010 13th International Conference on Information Fusion*. ; 2010:1-8.

172. Boukouvala F, Muzzio FJ, Ierapetritou MG. Predictive modeling of pharmaceutical processes with missing and noisy data. *AIChE J*. 2010;56:2860-2872.

173. Gomero B. Latin hypercube sampling and partial rank correlation coefficient. In Master Thesis. University of Tennessee, Knoxville; 2012:

174. Iooss B, Lemaître P. A review on global sensitivity analysis methods. In Meloni C, Dellino G, Uncertainty Management in Simulation-Optimization of Complex Systems. New York: Springer US; 2015:101-122.

175. Collinson S, Heffernan JM. Modelling the effects of media during an influenza epidemic. *BMC Public Health*. 2014;14:376.

176. Blower SM, Dowlatabadi H. Sensitivity and uncertainty analysis of complex models of disease transmission: An HIV model, as an example. *Int Stat Rev*. 1994;62:229-243.

177. IDEFØ – Function Modeling Method – IDEF.http://www.idef.com/idefo-function_modeling_method/. Accessed May 25, 2018

Appendix

A Appendix to Chapter 3

A.1 Confusion matrix

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{C}_{00} & \boldsymbol{C}_{10} \\ \boldsymbol{C}_{01} & \boldsymbol{C}_{11} \end{bmatrix}, \tag{A.1}$$

where:

C_{00} ≡ Number of times the process-class was correctly predicted

C_{11} ≡ Number of times the noise-class was correctly predicted

C_{10} ≡ Number of times the process-class was wrongly predicted

C_{01} ≡ Number of times the noise-class was wrongly predicted

A.2 Additional figures

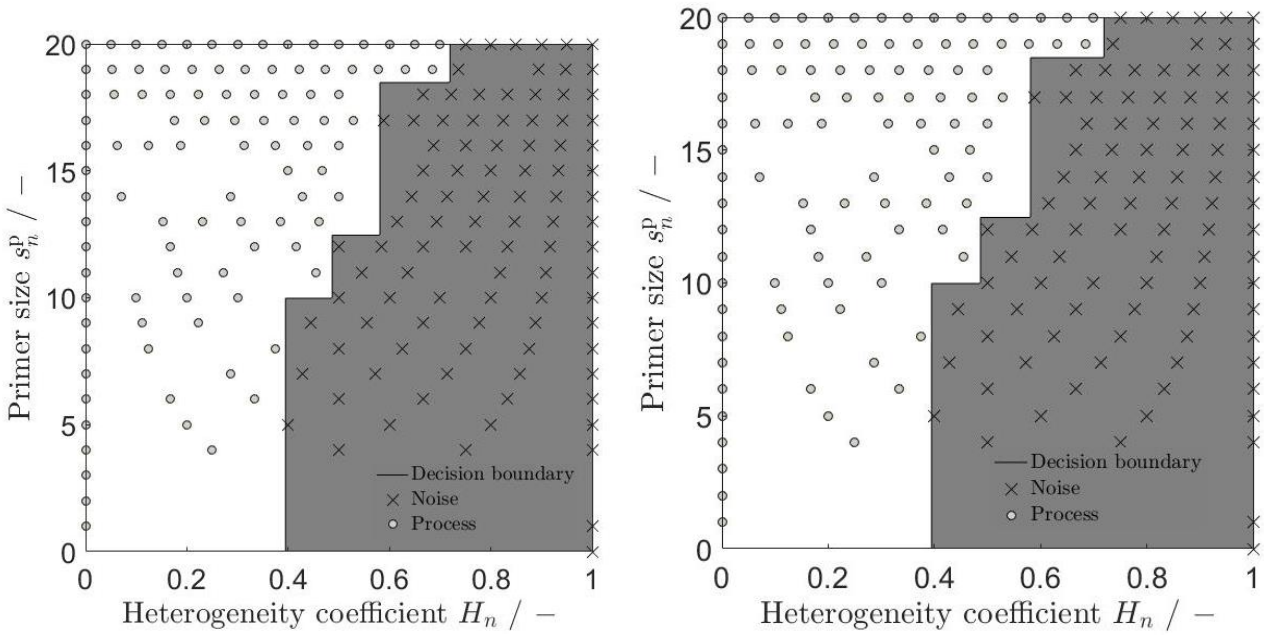


Figure. A.1 Classification results of the datasets $\boldsymbol{D_1}$**-2** (top) and $\boldsymbol{D_1}$**-3** (bottom)

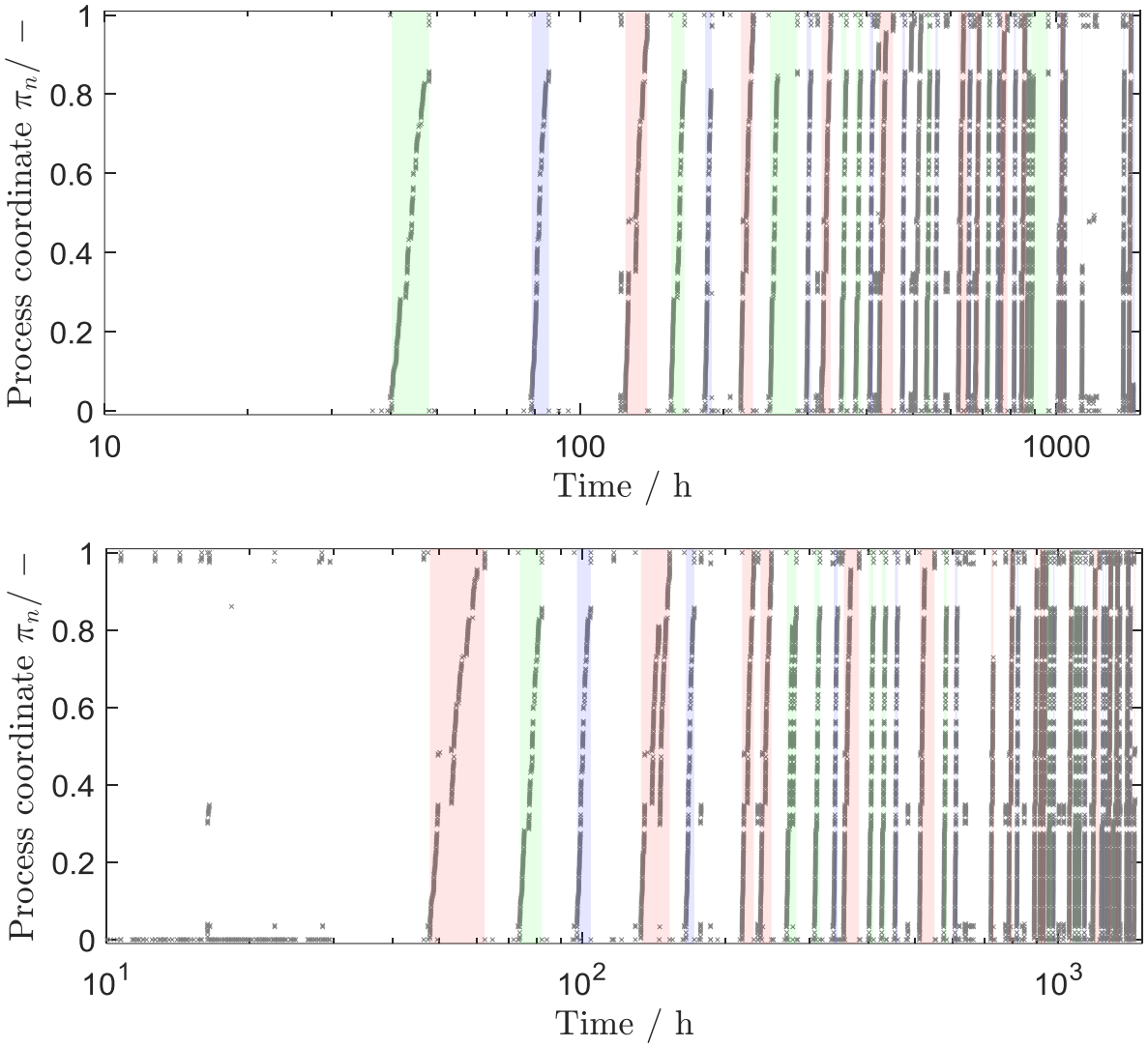


Figure A.2 Result of clustering on dataset ***D*-2** (top) and ***D*-3** (bottom)

A.3 Noise labeling algorithm

The Matlab function used for identifying the alarms, remedy and repetitions in the process output is presented below.

```
%% ”Cluster_list” is a cell containing task ID vectors as strings and execution times
%% vectors of task for each clusters
%% “Remedy_list” string vector containing the task ID of remedy operations
%% “Alarm_list” string vector containing the task ID of alarm operations
%% “Process_recipe” string vector containing the process sequence of task ID
%% “Alarm_position” Cell containing vectors with the position of each alarm
%% “Failing_taskID” Cell containing string vectors with the position of each alarm
%% “Repetition_position” Cell containing vectors with the position of the repeated tasks

function       [Alarm_position,       Failing_taskID,       Repetition_position]       =
failure_analysis_function(Remedy_list, Cluster_list, Alarm_list, Process_recipe)

% Pre-allocation
Alarm_position = cell(length(Cluster_list),1); %Position of the alarms with respect to n
Failure_presence = cell(length(Cluster_list),1); %Binary vector if =1 Failure exists
Failing_taskID = cell(length(Cluster_list),1); %List of ID of the failed tasks

for k=1:length(Cluster_list)

 if sum(ismember(Cluster_list {k,1}, Alarm _list)) ~= 0

  %% Identify Alarms
  Alarm_position{k,1} =find(ismember(Cluster_list {k,1}, Alarm_list));

  % Pre-allocation of the single cells
  Failing_taskID{k,1} = string(zeros(length(Alarm_position{k}),1));

  %% Identify failing tasks IDSs (the task before the alarm)
  for j =1:length(Alarm_position{k})
   Failing_taskID{k}(j) = Cluster_list {k,1}(Alarm_position{k}(j)-1);
  end

  %% Repetition recognition
  % Pre-allocation
  for j =1:length(Failing_taskID{k})
   Repetition_position{k,j}=[];
  end

  for j =1:length(Failing_taskID{k})
   i=Alarm_position{k}(j);
   T = 0;

   % Position of the failure in the recipe (B)
   [m,B,n] = intersect(Process_recipe,Failing_taskID{k}(j));
   while  B ~= T
    if Failing_taskID{k}(j) == string('Remedy)
     [m,T,n] = intersect(Process_recipe, Cluster_list{k,1}(i))
     Repetition_position{k,j} = [Repetition_position{k,j}(:,1); i];
    else
     if ~ismember(Cluster_list{k,1}(i), Alarm _list) &&
        ~ismember(Cluster_list{k,1}(i), Remedy_list)
      [m,T,n] = intersect(Process_recipe, Cluster_list{k,1}(i));
      Repetition_position{k,j} = [Repetition_position{k,j}(:,1); i];
     else
      Repetition_position{k,j} = [Repetition_position{k,j}; 0];
     end
    end
```

```
    i=i+1;
     if  i >= length(Cluster_list{k})
      T=B;
     end
    end
   end

   % Correct repetition sequence and recognize nested/consecutive failures
   for i =1:length(Failing_taskID{k})
    for j =i:length(Failing_taskID{k})
     [y,ia{i,j},ib]=intersect(Repetition_position{k,i},Repetition_position{k,j});
    end
   end

   % Pre-allocation set the position equal to 0 for all nested repetitions
   for i =1:length(Failing_taskID{k})
    for j =i:length(Failing_taskID{k})
     if i<j && sum(ia{i,j})~=0
      for t=1:length(ia{i,j})
       Repetition_position{k,j}(ia{i,j}(t))=0;
      end
     end
    end
   end

   for j =1:length(Failing_taskID{k})
    if sum(ismember(Repetition_position{k,j},0))~=0
     Repetition_position{k,j}(Repetition_position{k,j}(:,1)==0)=[];
    end
   end

   display('Repetition sequences were determined')

   % Recognition of remedy, which consist of the failure itself and the lag before
     restarting the process

   for j = 1:length(Alarm_position{k})
    if ~isempty(Repetition_position{k,j})
     Remedy_position{k,j} = Alarm_position{k}(j):Repetition_position{k,j}(1,1)-1;
    else
     Remedy_position{k,j} = [];
    end
   end
  end

  display('Remedy sequences were determined')

 end
end
```

A.4 Detailed algorithm performance results

Table A.1 Detailed results of data preprocessing of ***D***-1

Batch	Correctly identified	Batch deviation / %	Number of misclassification	Comment
1	Yes	0	0	
2	Yes	0	0	
3	Yes	0	0	
4	Yes	0	0	
5	Yes	0	0	
6	Yes	0	0	
7	Yes	0	0	
8	Yes	0	0	
9	Yes	0	0	
10	Yes	0	0	
11	Yes	0	0	
12	Yes	-6.4	10	Erroneously classified as noise
13	Yes	0	0	
14	Yes	0	0	
15	Yes	77	10	
16	Yes	0	0	
17	Yes	0	0	
18	Yes	510	138	Batch was not concluded following the recipe, no eETS (outlier)
19	Yes	0	0	
20	Yes	200	70	Misclassified noise
21	Yes	0	0	
22	Yes	0	0	
23	Yes	0	0	
24	Yes	0	0	
25	Yes	132	32	Misclassified noise
26	No	-	-	Partial batch
27	No	-	-	Partial batch
28	Yes	-15	18	Erroneously classified as noise
29	Yes	0	0	
30	Yes	189	14	Misclassified noise
31	Yes	0	0	
32	Yes	201	57	
33	Yes	0	0	
34	Yes	0	0	
35	Yes	460	62	Misclassified noise
36	Yes	0	0	
37	Yes	3.4	21	Misclassified noise
38	Yes	0	0	
39	Yes	0	0	

Table A.2 Detailed results of the data preprocessing of ***D*-2**

Batch	Correctly identified	Time deviation / %	Number of misclassification	Comment
1	Yes	0	0	
2	Yes	0	0	
3	Yes	0	0	
4	Yes	0	0	
5	Yes	-8.4	15	Erroneously classified as noise
6	Yes	0	0	
7	Yes	0	0	
8	Yes	0	0	
9	Yes	0	0	
10	Yes	0	0	
11	Yes	0	0	
12	Yes	0	0	
13	Yes	0	0	
14	Yes	0	0	
15	No	-	0	Partial batch
16	No	-	0	Partial batch
17	Yes	-19	31	Erroneously classified as noise
18	Yes	0	0	
19	Yes	0	0	
20	Yes	0	0	
21	Yes	0	0	
22	Yes	0	0	
23	Yes	31	5	Misclassified noise
24	Yes	0	0	
25	Yes	0	0	
26	Yes	0	0	
27	Yes	0	0	
28	Yes	0	0	
29	Yes	-25	12	Erroneously classified as noise
30	Yes	0	0	
31	Yes	0	0	
32	Yes	-83	12	Batch was not concluded following the recipe, no eETS (outlier)
33	Yes	0	0	
34	Yes	0	0	
35	Yes	0	0	Not a commercial batch (excluded)

Table A.3 Detailed results of the data pre-processing of ***D*-3**

Batch	Correctly identified	Time deviation / %	Number of misclassification	Comment
1	Yes	0	0	
2	Yes	0	0	
3	Yes	0	0	
4	Yes	0	0	
5	Yes	0	0	
6	Yes	1.0	6	Misclassified noise
7	Yes	0	0	
8	Yes	0	0	
9	Yes	0	0	
10	Yes	0	0	
11	Yes	0	0	
12	Yes	0	0	
13	Yes	0	0	
14	Yes	0	0	
15	Yes	0	0	
16	Yes	0	0	
17	Yes	0	0	
18	Yes	-5.4	25	Erroneously classified as noise
19	Yes	0	0	
20	Yes	0	0	
21	Yes	-19	35	
22	Yes	0	0	
23	Yes	-21	50	Erroneously classified as noise
24	Yes	0	0	
25	Yes	0	0	
26	Yes	0	0	
27	Yes	0	0	
28	Yes	2.3	10	Misclassified noise
29	Yes	0	0	
30	Yes	0	0	
31	No	-	30	Erroneously classified as noise, Partial batch
32	No	-	-	Partial batch
33	Yes	0	0	
34	Yes	0	0	
35	Yes	-0.5	8	Erroneously classified as noise
36	Yes	0	0	
37	Yes	0	0	
38	Yes	0	0	
39	Yes	0	0	
40	Yes	0	0	

B Appendix to Chapter 4

B.1 Algorithm for calculating c_k

1. *Initialize the matrix* $\boldsymbol{F}$ *with size* $[K \times 1]$.
2. *Sample a random number from* $B(f_k, p_k)$ *for each* $\boldsymbol{F}_{k,1}$.
3. *Return* $\boldsymbol{F}$.

Failure in task k in layer j is represented by an element equal to 1 in a binary-element failure matrix $\boldsymbol{F}$, the element of which is $\boldsymbol{F}_{k,j}$, a portion of the entire Bayesian network of failures. For task k the matrix at the first layer $\boldsymbol{F}_{*,1}$ is created by sampling from the Bernoulli probability distributions $B(f_k, p_k)$. If all elements of $\boldsymbol{F}_{*,1}$ are equal to 0, no failure occurred and the algorithm is terminated. Otherwise, the algorithm continues as follows.

1. *Find k for non-zero elements of* $\boldsymbol{F}_{k,j}$.
2. *Update* $j = j + 1$.
3. *For each k and each non-zero element of* $\boldsymbol{R}_{*,k}$, *sample a random value from* $B(f_k, p_k)$
4. *Update* $\boldsymbol{F}_{*,j}$.
5. *If* $\sum_k \boldsymbol{F}_{k,j} \neq 0$ *iterate from 1.*
6. *Return* $\boldsymbol{F}$.

As long as the sum over the k elements in the last column j of $\boldsymbol{F}$ is nonzero, a new column $(j + 1)$ is added to matrix $\boldsymbol{F}$, and new values are sampled from $B(f_k, p_k)$. In the new column $(j + 1)$, the values are sampled only for the task k' that has to be repeated in the case of failure in task k, namely if $\boldsymbol{R}_{k',k}^{\mathrm{CIP/SIP}} = 1$, whereas for other tasks a value of 0 is assigned. If the sum over the k elements in the last column j of $\boldsymbol{F}$ is equal to 0, the algorithm is terminated. The parameter c_k^{rep} is calculated by counting the failures that occurred in task k, shown in row k of matrix $\boldsymbol{F}$.

$$c_k^{\mathrm{rep}} = \sum_j \boldsymbol{F}_{k,j}$$

B.2 PRCC calculation

The PRCC is calculated for every changeable parameter by LHS–MCS.[173–176] The stratified sample matrix $\boldsymbol{X}$ of the changeable parameters and the vector $\boldsymbol{KPI}$ are paired. The samples within each changeable parameter are

ranked by assigning the ranking values $r_{k,n} \in [1, N]$ (see Eq. (B1)), where the highest rank is assigned to the highest value in every column.

$$|\boldsymbol{X} \quad \boldsymbol{KPI}| = \begin{vmatrix} r_{1,1} & r_{2,1} & \cdots & r_{k,1} & r_{K,1} & r_{K+1,1} \\ r_{1,n} & r_{2,n} & \cdots & r_{k,n} & r_{K,n} & r_{K+1,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ r_{1,N} & r_{2,N} & \cdots & r_{k,N} & r_{K,N} & r_{K+1,N} \end{vmatrix} \tag{B1}$$

The ranking matrix of size $[N,\ K+1]$ shown in equation A1 is used to calculate the intermediate symmetric matrix $\boldsymbol{S}$ as shown by equation B2.

$$\boldsymbol{S} = \boldsymbol{C^{-1}} = \left[c_{k',k''}\right]^{-\mathbf{1}} = \frac{\sum_{n=1}^{N}\left(r_{k',n}-\mu\right)\left(r_{k'',n}-\mu\right)}{\sqrt{\sum_{n=1}^{N}\left(r_{k',n}-\mu\right)^2 \sum_{s=1}^{N_i}\left(r_{k'',s}-\mu\right)^2}} \tag{B2}$$

$$k', k'' = 1,2,\ldots,K+1$$

The parameter μ is the average rank of the sample and is equal to $(N+1)/2$. The matrix $\boldsymbol{C}$ is subsequently used to calculate the PRCC ρ_k, as shown in equation S3.

$$\rho_k = \frac{-c_{k',K+1}}{\sqrt{c_{k',k'}c_{K+1,K+1}}} \tag{B3}$$

The significance of "ρ_k is different from 0" is tested by a t-test,[176] calculating the t_k-value as follows.

$$t_k = \rho_k \sqrt{\frac{N-2}{1-\rho_k}} \tag{B4}$$

The PRCC approach has various advantages, such as simplicity and low computational effort. Additionally, it can also evaluate nonlinear and stochastic models; however, it is only applicable if the relationship between input data and output data, here represented by $\boldsymbol{X}$ and $\boldsymbol{KPI}$, is monotonic.

B.3 Additional figure

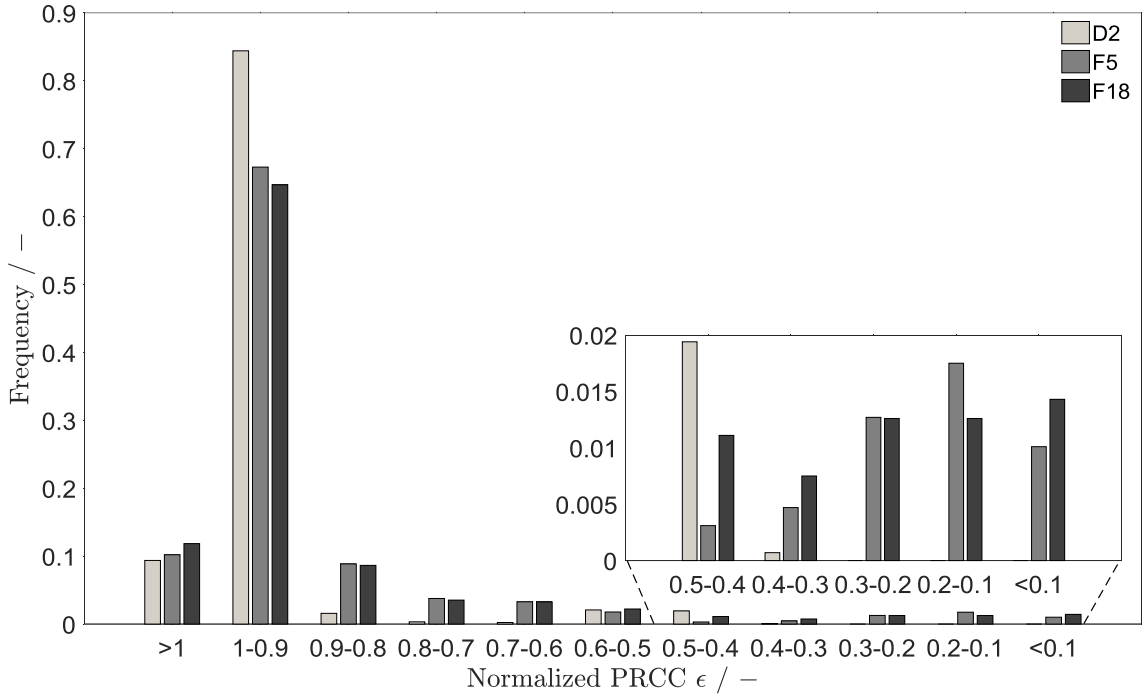


Figure B.1 RNE analysis results for the tasks *D2*, *F5*, and *F18* belonging to *intra*-CIPSIP in the case study.

B.4 LSS assessment

An analysis using LSS was conducted in the case study for comparative purposes. The process performance model shown in equation B5 is analogous to the one used in the method (see Eq. (4.8)), but the model considers the annual process run time neglecting repetitions and failures.

$$T_{\mathrm{CIP/SIP}}^{\mathrm{annual}} = \sum_{k=1}^{K} t_k^{\mathrm{annual}} \tag{B5}$$

The variables $T_{\mathrm{CIP/SIP}}^{\mathrm{annual}}$ and t_k^{annual} represent the total annual run times invested in CIP/SIP and task k, respectively. The $ratio_k^{\mathrm{LSS}}$ is constructed on the basis of the partial contribution of every task k to the total annual run time, as shown in equation B6.

$$ratio_k^{\mathrm{LSS}} = \frac{t_k^{\mathrm{annual}}}{T_{\mathrm{CIP/SIP}}^{\mathrm{annual}}} \tag{B6}$$

The ratio is traditionally represented by a so-called Pareto chart. **Figure B.2** shows a Pareto chart where $ratio_k^{\mathrm{LSS}}$ is listed from the highest to the lowest for the two processes, *post*-CIP/SIP (left) and *intra*-CIP/SIP (right). The *y*-axis on the right and the black line show the cumulative yearly contribution.

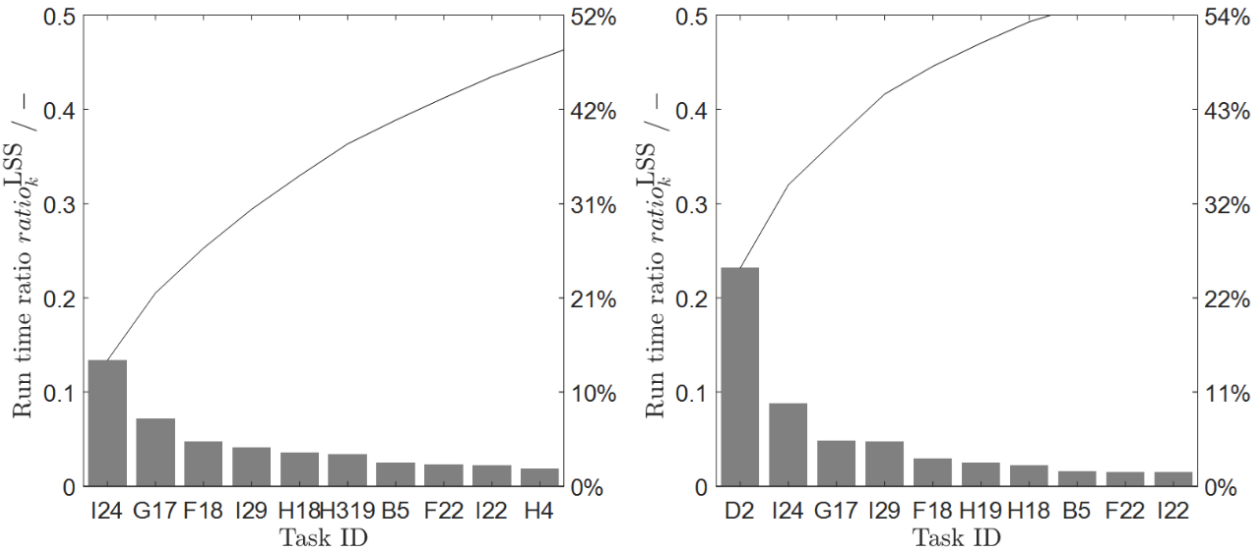


Figure B.2 Pareto chart showing the annual run time contribution of each task, for *post*-CIPSIP (left) and *intra*-CIP/SIP (right).

C Appendix to Chapter 5

C.1 Contingency table

Table C.1 Generic graphical representation showing the difference between operational disturbance (red) and noise (gray)

		Real failure class	
		Successful	*About to fail*
Predicted failure class	*Successful*	True positive	False positive (Type I error)
	About to fail	False negative (Type II error)	True negative

C.2 Block specific sensitivity analysis

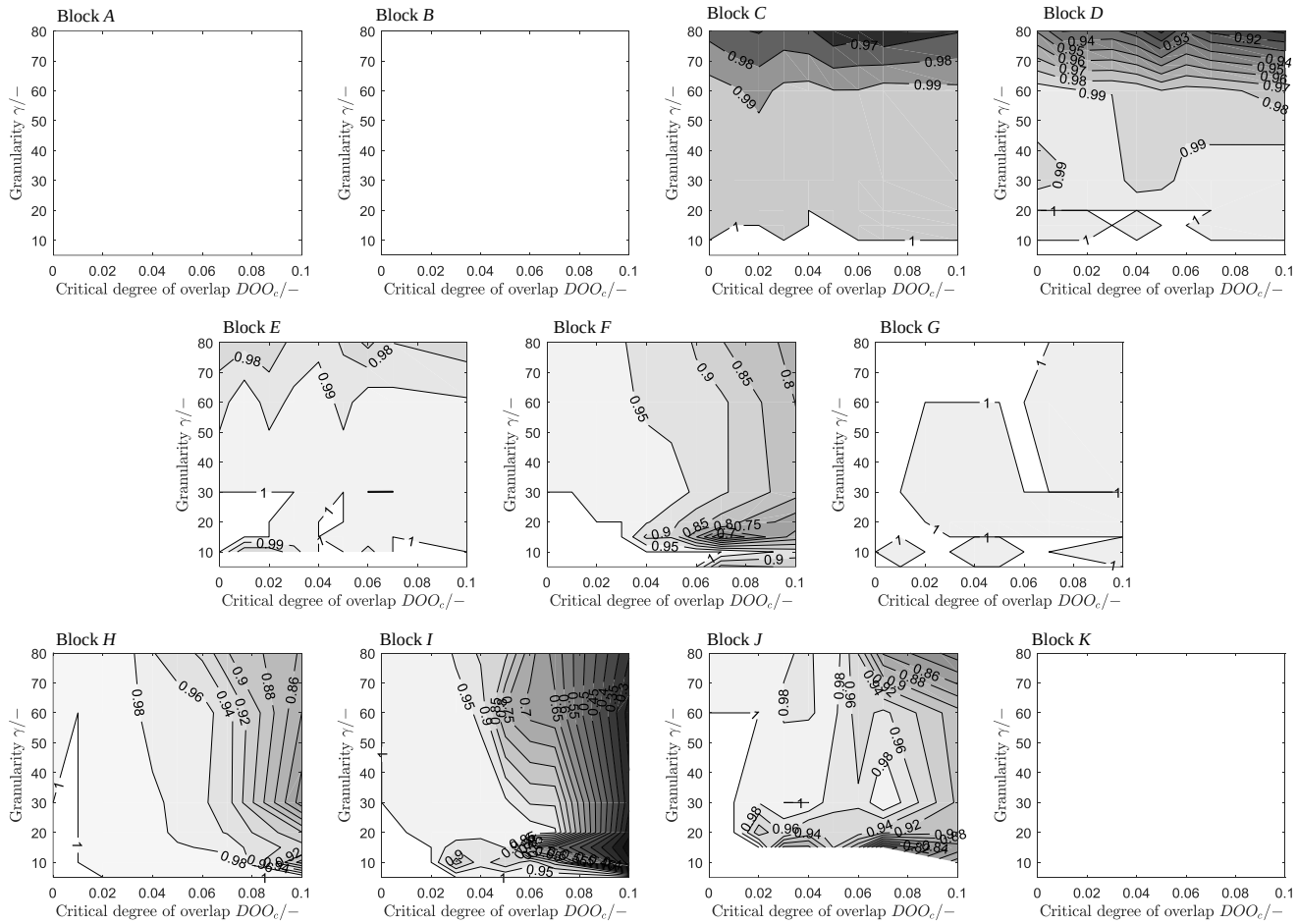


Figure C.2.1 Block-specific sensitivity analysis, the models are trained on an 85 batches dataset.

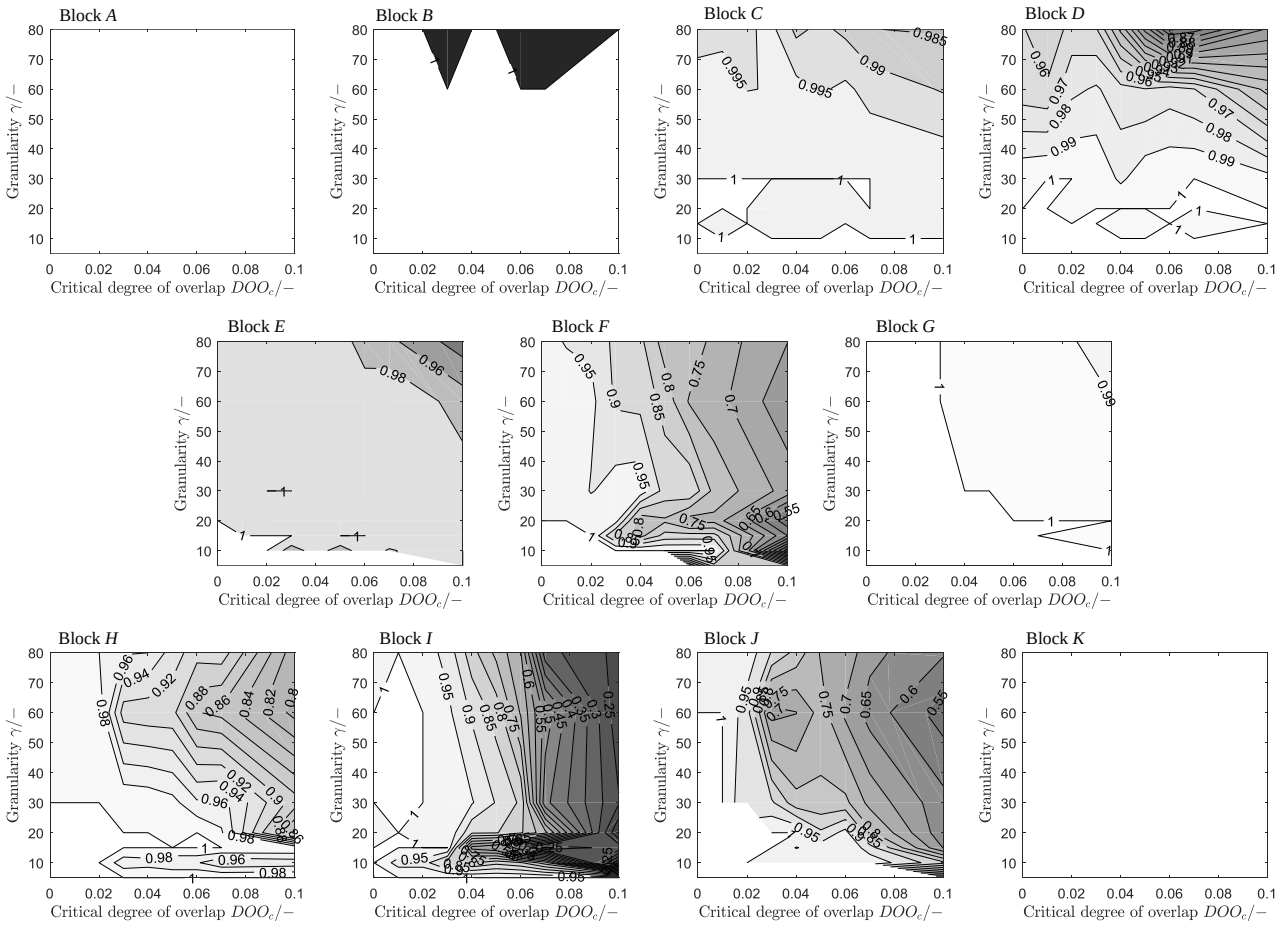


Figure C.2.2 Block-specific sensitivity analysis, the models are trained on a 85 batches dataset

C.3 Training details

Table C.3.1 Training performance (NPV) of the block-specific classification models

(N.A. appears whenever the training dataset is entirely classified as *successful*)

Block	Conservative scenario	Risky scenario
A	1	0.99
B	1	1
C	1	1
D	1	0.99
E	1	1
F	1	0.86
G	1	0.99
H	1	0.87
I	N.A.	0.75
J	N.A.	0.13
K	N.A.	0.23
mean	1	0.81

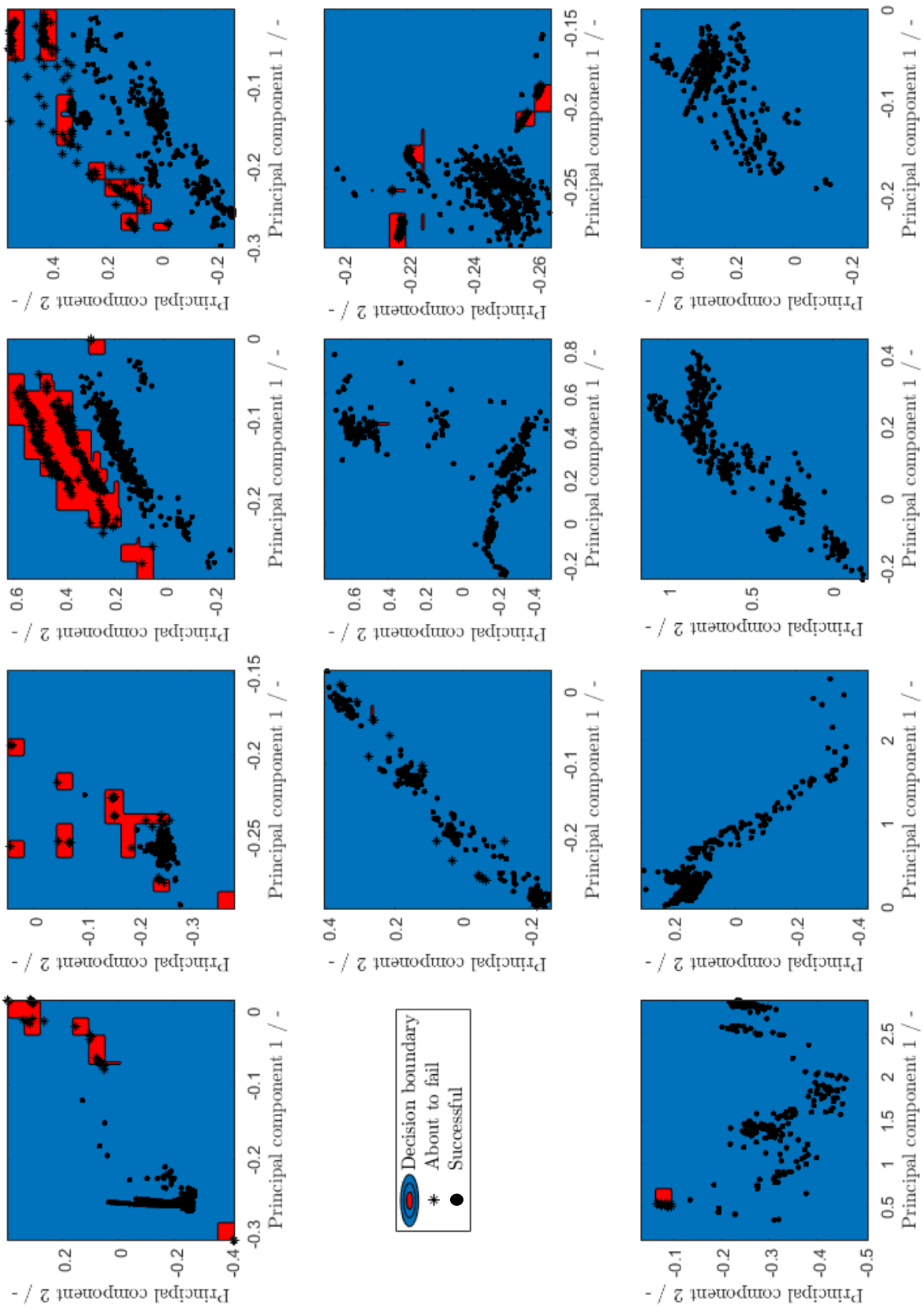


Figure C.3.2 Calculated failure decision boundary for each process block; *conservative* scenario with training data.

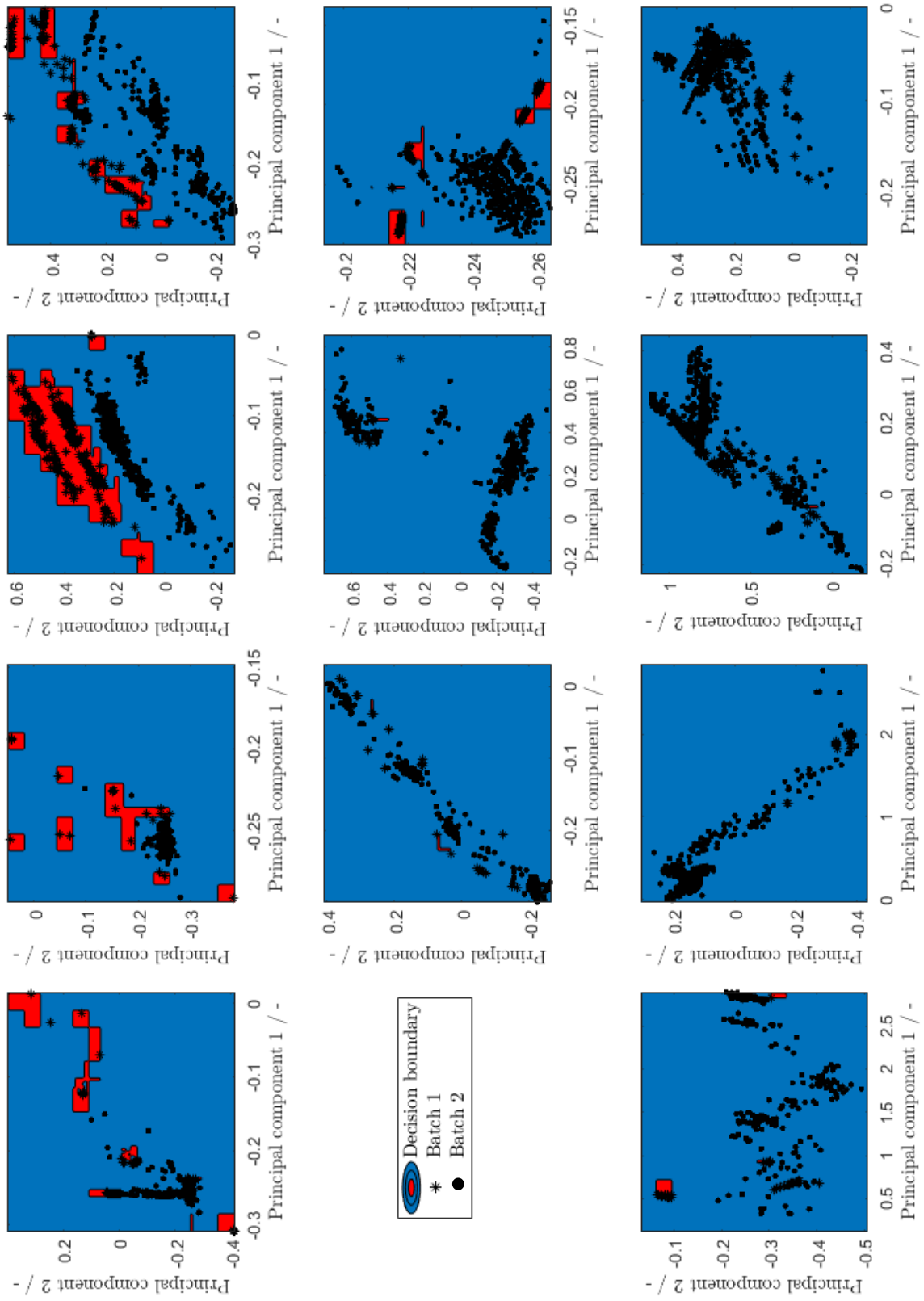


Figure C.3.2 Calculated failure decision boundary for each process block; *risky* scenario with training data.

C.4 Evolution of the process space

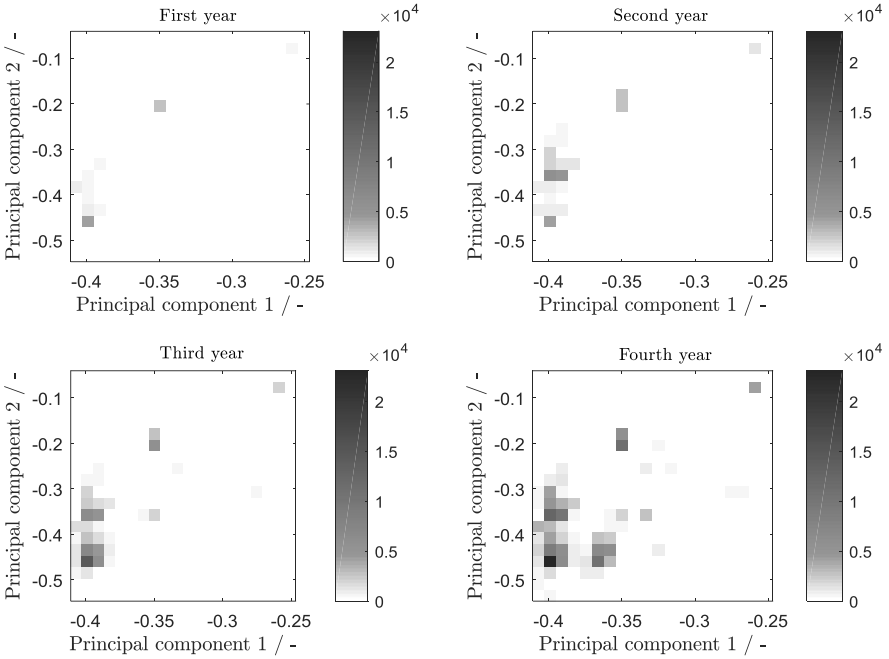


Figure C.4.1 Process space evolution for block *A*

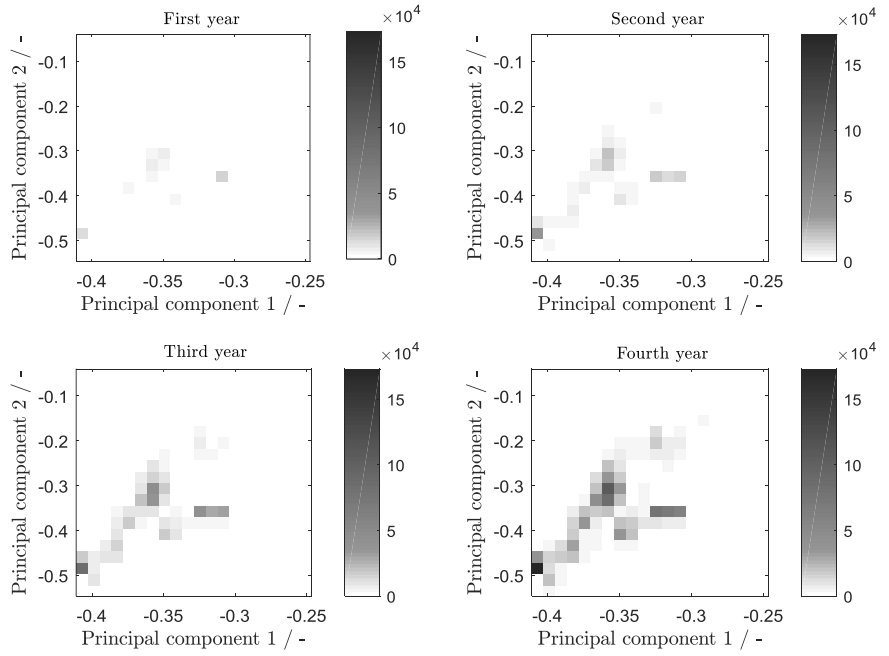


Figure C.4.2 Process space evolution for block *B*

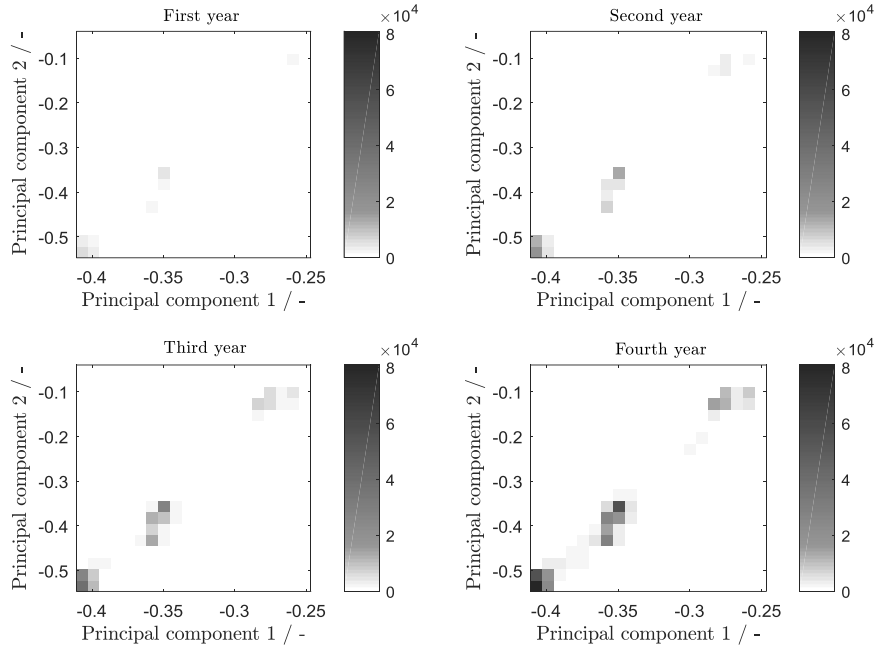


Figure C.4.3 Process space evolution for block *D*

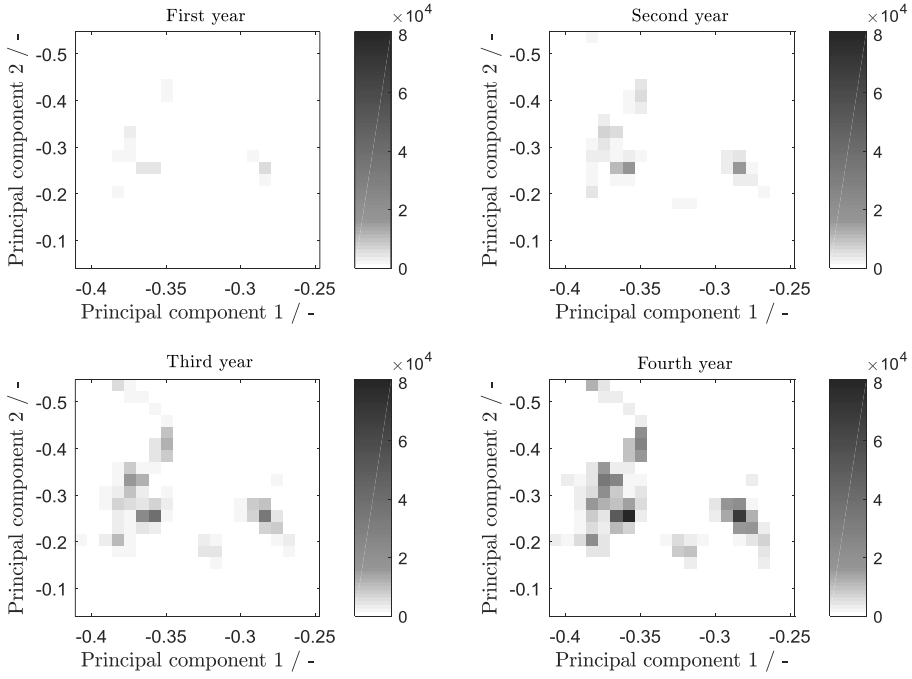


Figure C.4.4 Process space evolution for block *E*

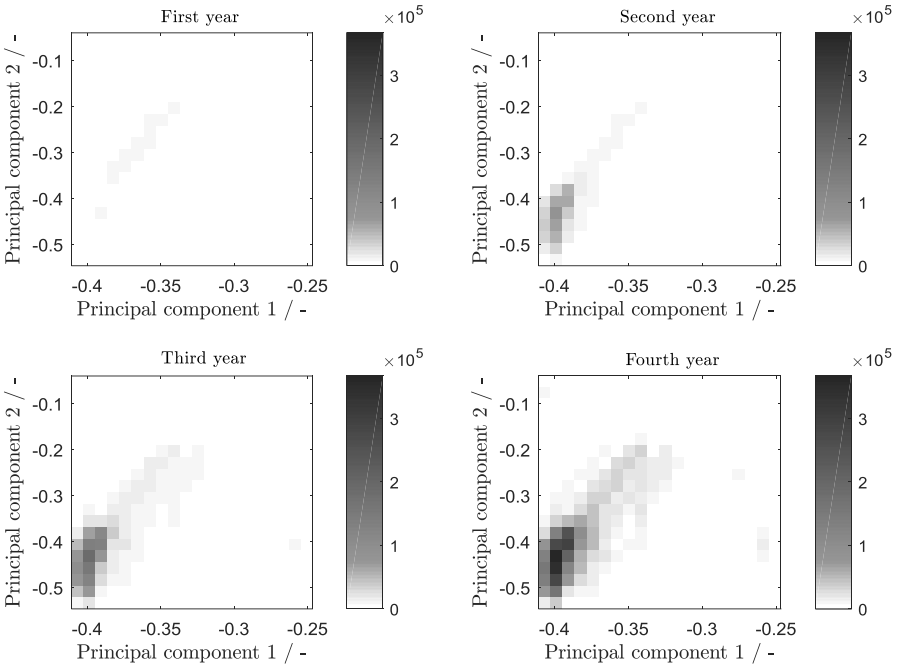


Figure C.4.5 Process space evolution for block *F*

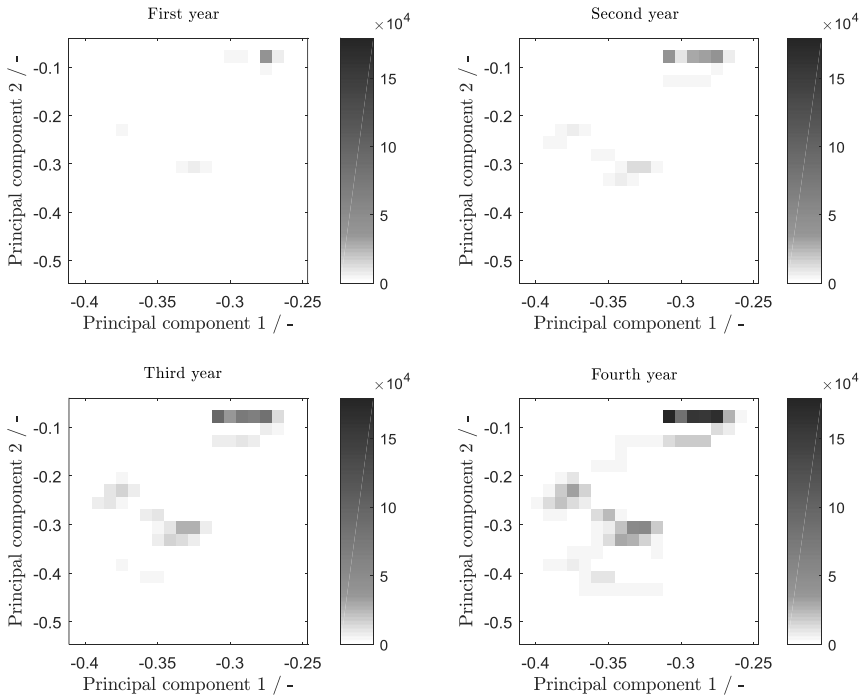


Figure C.4.6 Process space evolution for block *G*

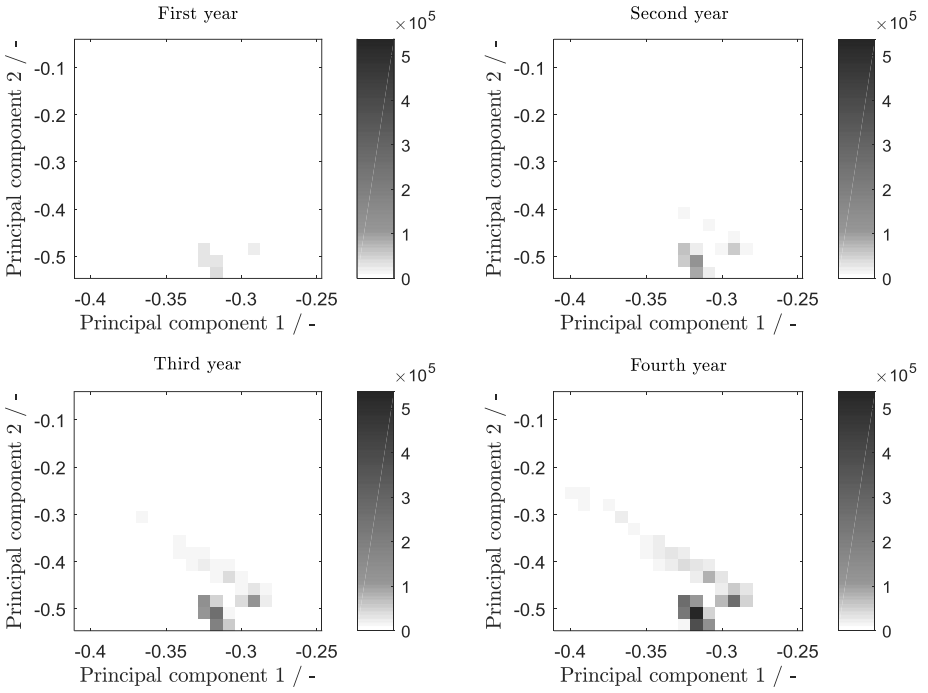


Figure C.4.7 Process space evolution for block *H*

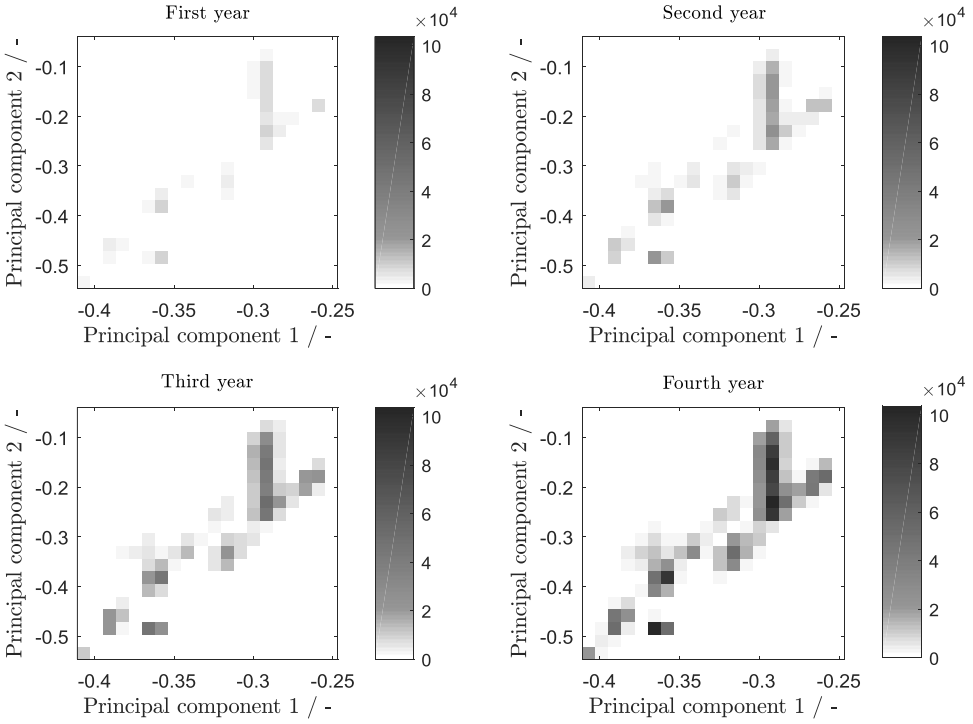


Figure C.4.8 Process space evolution for block *I*

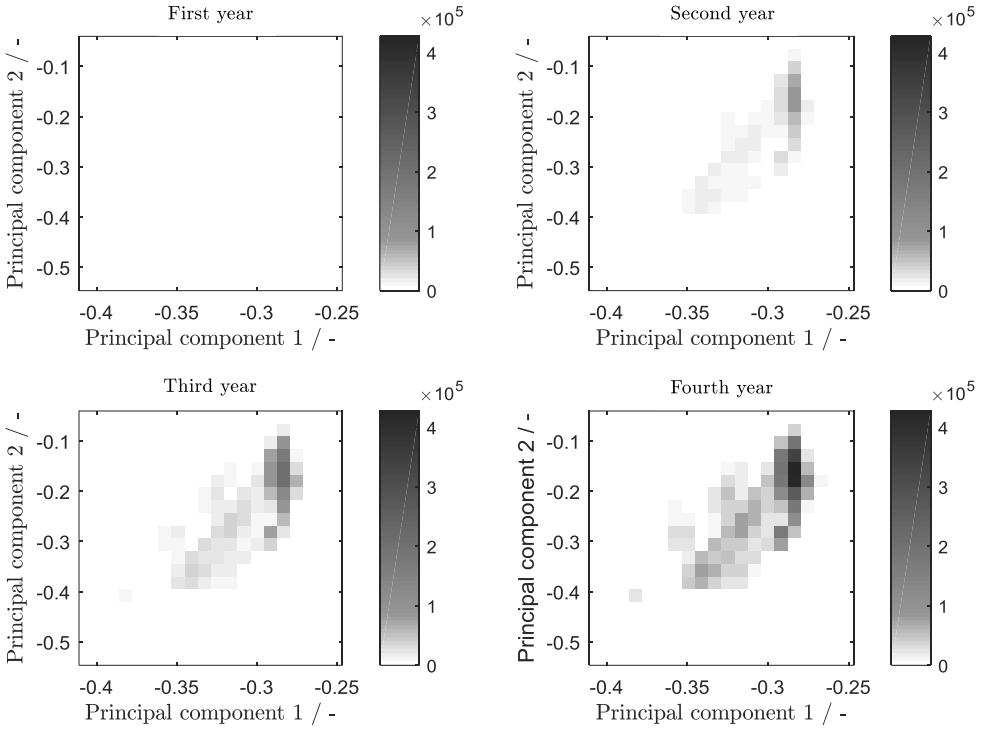


Figure C.4.9 Process space evolution for block *J*

Table C.4.1 Summary table of the results of the simulation of the intelligent self-training algorithm for the *conservative* scenario

Batch ID	Detection time [h]	Status	Duration of the batch [h]	Time saved [h]
1	-	For training	0.76	-
2	-	For training	1.38	-
3	-	For training	0.04	-
4	-	For training	17.36	-
5	-	For training	0.03	-
6	-	For training	0.19	-
7	-	For training	12.43	-
8	-	For training	0.02	-
9	-	For training	12.39	-
10	-	For training	12.47	-
11	-	For training	52.67	-
12	-	For training	0.04	-
13	-	For training	12.60	-
14	-	For training	12.37	-
15	-	For training	12.41	-
Retraining ($\overline{\text{NPV}} = 1$)				
16	-	-	12.27	-
17	-	-	13.41	-
18	-	-	0.03	-

19	0.00	Lost	1.85	0.00
20	-	-	12.63	-
21	-	-	12.49	-
22	-	-	15.39	-
23	-	-	0.03	-
24	-	-	0.04	-
25	-	-	12.58	-
26	-	-	12.36	-
27	0.00	Lost	0.02	0.00
28	-	-	12.50	-
29	-	-	12.56	-
30	0.00	Lost	14.30	0.00
Retraining ($\overline{\text{NPV}} = 1$)				
31	0.00	Lost	21.37	0.00
32	-	-	4.70	-
33	-	-	0.20	-
34	-	-	0.99	-
35	-	-	5.68	-
36	-	-	4.51	-
37	-	-	4.76	-
38	-	-	1.37	-
39	-	-	10.61	-
40	-	-	6.18	-
41	-	-	0.70	-
43	0.00	Lost	0.14	0.00
44	-	-	13.03	-
Retraining ($\overline{\text{NPV}} = 1$)				
45	-	-	1.87	-
46	0.00	Lost	49.54	0.00
47	8.73	Salvaged	10.29	1.57
48	1.04	Lost	36.50	-1.04
49	0.00	Lost	7.00	0.00
50	-	-	0.59	-
51	0.00	Lost	0.09	0.00
52	-	-	2.03	-
53	-	-	1.79	-
54	-	-	2.48	-
56	-	-	0.04	-
57	-	-	52.33	-
58	-	-	0.19	-
59	-	-	3.76	-
Retraining ($\overline{\text{NPV}} = 1$)				
60	-	-	0.01	-
61	0.00	Lost	12.53	0.00
62	-	-	0.03	-

63	0.00	Lost	13.58	0.00
64	0.00	Lost	0.01	0.00
65	-	-	12.90	-
66	-	-	0.02	-
67	-	-	0.02	-
68	-	-	3.12	-
69	-	-	2.40	-
70	-	-	2.32	-
71	-	-	13.55	-
72	0.00	Lost	0.03	0.00
73	0.00	Salvaged	1.80	1.80
74	0.02	Salvaged	0.69	0.67
Retraining ($\overline{\text{NPV}} = 1$)				
75	-	-	1.93	-
76	-	-	0.00	-
77	-	-	7.16	-
78	-	-	29.66	-
79	-	-	12.40	-
80	-	-	37.53	-
81	-	-	12.47	-
82	0.01	Salvaged	6.63	6.63
83	1.83	Salvaged	2.59	0.76
84	-	-	17.52	-
85	-	-	1.71	-
86	-	-	0.20	-
87	-	-	0.73	-
88	-	-	12.67	-
89	-	-	0.07	-
Retraining ($\overline{\text{NPV}} = 1$)				
90	-	-	0.05	-
91	-	-	0.27	-
92	-	-	12.26	-
93	-	-	0.00	-
94	-	-	1.24	-
95	-	-	0.29	-
96	-	-	0.07	-
97	-	-	0.42	-
98	-	-	12.82	-
99	-	-	7.13	-
100	-	-	0.34	-
101	1.04	Salvaged	17.76	16.73
102	0.00	Lost	12.49	0.00
Retraining ($\overline{\text{NPV}} = 1$)				
103	-	-	0.01	-
104	12.73	Lost	15.77	-12.73

105	-	-	4.06	-
106	1.26	Lost	17.04	-1.26
107	-	-	10.78	-
108	-	-	0.02	-
109	1.56	Salvaged	2.82	1.26
110	-	-	0.18	-
111	-	-	0.14	-
112	-	-	12.89	-
113	0.00	Lost	0.11	0.00
114	-	-	27.95	-
115	-	-	37.74	-
Retraining ($\overline{\text{NPV}} = 1$)				
116	-	-	0.02	-
117	-	-	0.04	-
118	-	-	7.06	-
119	-	-	2.38	-
120	-	-	12.51	-
121	0.00	Lost	0.07	0.00
122	-	-	0.03	-
123	-	-	8.40	-
124	-	-	27.45	-
125	-	-	32.45	-
126	-	-	1.21	-
127	-	-	14.02	-
128	-	-	14.14	-
Retraining ($\overline{\text{NPV}} = 1$)				
129	-	-	0.02	-
130	-	-	0.29	-
131	-	-	12.58	-
132	-	-	12.57	-
133	-	-	2.10	-
134	-	-	1.40	-
135	0.00	Lost	0.14	0.00
136	-	-	0.49	-
137	0.00	Lost	0.03	0.00
Retraining ($\overline{\text{NPV}} = 1$)				
138	0.06	Salvaged	0.16	0.10
139	-	-	12.40	-
140	-	-	0.02	-
141	-	-	0.01	-
142	-	-	30.19	-
143	-	-	0.01	-
144	6.03	Lost	37.88	-6.03
145	5.71	Lost	12.70	-5.71
146	-	-	0.02	-

147	-	-	1.61	-
148	-	-	0.51	-
Retraining ($\overline{\text{NPV}} = 1$)				
149	0.01	Salvaged	1.50	1.49
150	-	-	0.80	-
151	-	-	14.26	-
152	-	-	13.38	-
153	-	-	0.00	-
154	-	-	1.37	-
155	-	-	30.93	-
156	-	-	14.56	-
157	-	-	2.96	-
158	-	-	0.01	-
159	-	-	0.04	-
160	-	-	2.46	-
161	-	-	0.03	-
162	-	-	0.97	-
Retraining ($\overline{\text{NPV}} = 1$)				
163	-	-	1.91	-
164	-	-	14.17	-
165	7.62	Lost	13.59	-7.62
166	0.00	Lost	14.58	0.00
167	-	-	12.30	-
168	-	-	0.03	-
169	-	-	2.71	-
170	-	-	0.05	-
171	-	-	0.01	-
172	0.00	Lost	0.07	0.00
173	-	-	0.03	-
174	-	-	0.86	-
175	-	-	15.22	-
176	-	-	1.31	-
177	-	-	12.27	-
Retraining ($\overline{\text{NPV}} = 1$)				
178	-	-	6.10	-
179	-	-	14.67	-
180	-	-	12.64	-
181	-	-	0.04	-
182	-	-	12.74	-
183	5.63	Lost	26.16	-5.63
184	2.46	Lost	13.70	-2.46
185	-	-	0.28	-
186	6.88	Lost	13.43	-6.88
187	6.18	Lost	17.20	-6.18
188	-	-	1.56	0.00

189	6.82	Lost	13.49	-6.82
190	8.25	Lost	15.29	-8.25
191	-	-	1.41	-
192	-	-	1.42	-
Retraining ($\overline{\text{NPV}} = 1$)				
193	-	-	14.44	-
194	-	-	1.39	-
195	-	-	5.27	-
196	-	-	13.25	-
197	-	-	1.38	-
198	0.05	Salvaged	0.34	0.30
199	-	-	1.36	-
200	-	-	12.80	-
201	0.02	Salvaged	1.40	1.38
202	-	-	1.76	-
203	-	-	0.11	-
204	-	-	0.01	-
205	-	-	1.51	-
206	-	-	1.44	-
Retraining ($\overline{\text{NPV}} = 1$)				
207	-	-	1.44	-
208	-	-	1.43	-
209	-	-	0.02	-
210	0.04	Salvaged	1.40	1.35
211	-	-	1.40	-
212	-	-	0.05	-
213	-	-	1.44	-
214	-	-	1.45	-
215	-	-	0.06	-
216	-	-	0.01	-
217	-	-	1.47	-
218	-	-	15.30	-
219	-	-	0.34	-
220	-	-	1.54	-
Retraining ($\overline{\text{NPV}} = 1$)				
221	-	-	0.01	-
222	-	-	1.50	-
223	-	-	0.05	-
224	-	-	1.50	-
225	-	-	1.66	-
226	-	-	1.41	-
227	-	-	0.01	-
228	-	-	1.48	-
229	-	-	1.42	-
230	-	-	1.45	-

231	-	-	1.42	-
232	-	-	1.46	-
233	-	-	0.26	-
234	-	-	0.01	-
Retraining ($\overline{\text{NPV}} = 1$)				
235	-	-	0.01	-
236	-	-	2.80	-
237	-	-	2.19	-
238	-	-	37.76	-

Table C.4.2 Summary table of the results of the simulation of the intelligent self-training algorithm for the *risky* scenario

Batch ID	Detection time [h]	Status	Duration of the batch [h]	Time saved [h]
1	-	For training	0.76	-
2	-	For training	1.38	-
3	-	For training	0.04	-
4	-	For training	17.36	-
5	-	For training	0.03	-
6	-	For training	0.19	-
7	-	For training	12.43	-
8	-	For training	0.02	-
9	-	For training	12.39	-
10	-	For training	12.47	-
11	-	For training	52.67	-
12	-	For training	0.04	-
13	-	For training	12.60	-
14	-	For training	12.37	-
15	-	For training	12.41	-
Retraining ($\overline{\text{NPV}} = 0.88$)				
16	0.67	Lost	12.27	-0.67
17	0.57	Lost	13.41	-0.57
18	-	-	0.03	-
19	0.00	Salvaged	1.85	1.85
20	-	-	12.63	0.00
21	0.94	Lost	12.49	-0.94
22	1.00	Lost	15.39	-1.00
23	-	-	0.03	-
24	-	-	0.04	-
25	0.20	Lost	12.58	-0.20
26	0.62	Lost	12.36	-0.62
27	0.00	Lost	0.02	0.00
28	0.58	Lost	12.50	-0.58
29	0.59	Lost	12.56	-0.59
30	0.00	Lost	14.30	0.00

Retraining ($\overline{\text{NPV}} = 0.80$)				
31	0.00	Lost	21.37	0.00
32	-	-	4.70	-
33	-	-	0.20	-
34	0.65	Salvaged	0.99	0.34
35	0.68	Salvaged	5.68	5.00
36	0.60	Salvaged	4.51	3.92
37	0.29	Salvaged	4.76	4.47
38	-	-	1.37	-
49	1.02	Salvaged	10.61	9.60
50	-	-	6.18	-
51	-	-	0.00	-
52	-	-	0.70	-
53	0.00	Lost	0.14	0.00
54	-	-	13.03	-
Retraining ($\overline{\text{NPV}} = 0.81$)				
55	-	-	1.87	-
56	0.00	0.00	49.54	0.00
57	1.76	Salvaged	10.29	8.54
58	1.04	Lost	36.50	-1.04
59	0.00	Lost	7.00	0.00
60	-	-	0.59	-
61	0.00	Lost	0.09	0.00
62	-	-	2.03	-
63	-	-	1.79	-
64	-	-	0.00	-
65	-	-	2.48	-
66	-	-	0.04	-
67	6.01	Lost	52.33	-6.01
68	-	-	0.19	-
69	-	-	3.76	-
Retraining ($\overline{\text{NPV}} = 0.92$)				
70	-	-	0.01	-
71	0.00	Lost	12.53	0.00
72	-	-	0.03	-
73	0.00	Lost	13.58	0.00
74	0.00	Lost	0.01	0.00
75	-	-	12.90	-
76	-	-	0.02	-
77	-	-	0.02	-
78	1.30	Salvaged	3.12	1.82
79	-	-	2.40	-
80	-	-	2.32	-
81	0.00	Lost	13.55	0.00
82	-	-	0.03	-

83	0.00	Salvaged	1.80	1.80
84	0.02	Salvaged	0.69	0.67
Retraining ($\overline{\text{NPV}} = 0.80$)				
85	-	-	1.93	-
86	-	-	7.16	-
87	2.14	Lost	29.66	-2.14
88	-	-	12.40	-
89	-	-	37.53	-
90	-	-	12.47	-
91	0.01	Salvaged	6.63	6.63
92	1.83	Salvaged	2.59	0.76
93	-	-	17.52	-
94	1.18	Salvaged	1.71	0.54
95	-	-	0.20	-
96	-	-	0.73	-
97	0.26	Lost	12.67	-0.26
98	-	-	0.07	-
Retraining ($\overline{\text{NPV}} = 0.90$)				
99	-	-	0.05	-
100	-	-	0.27	-
101	-	-	12.26	-
102	-	-	0.00	-
103	-	-	1.24	-
104	-	-	0.29	-
105	-	-	0.07	-
106	-	-	0.42	-
107	3.03	Lost	12.82	-3.03
108	-	-	7.13	-
109	-	-	0.34	-
110	1.04	Salvaged	17.76	16.73
111	0.00	Lost	12.49	0.00
Retraining ($\overline{\text{NPV}} = 0.81$)				
112	-	-	0.01	-
113	2.93	Lost	15.77	-2.93
114	-	-	4.06	0.00
115	0.88	Lost	17.04	-0.88
116	-	-	10.78	-
117	-	-	0.02	-
118	1.56	Salvaged	2.82	1.26
119	-	-	0.18	-
120	-	-	0.14	-
121	2.04	Lost	12.89	-2.04
122	0.00	Lost	0.11	0.00
123	2.01	Lost	27.95	-2.01
124	2.00	Lost	37.74	-2.00

Retraining ($\overline{\text{NPV}} = 0.84$)				
125	-	-	0.02	-
126	-	-	0.04	-
127	1.98	Salvaged	7.06	5.08
128	-	-	2.38	-
129	2.08	Lost	12.51	-2.08
130	0.00	Lost	0.07	0.00
131	-	-	0.03	-
132	2.31	Salvaged	8.40	6.08
133	7.13	Lost	27.45	-7.13
134	2.42	Lost	32.45	-2.42
135	-	-	1.21	-
136	1.39	Lost	14.02	-1.39
137	6.69	Lost	14.14	-6.69
Retraining ($\overline{\text{NPV}} = 0.84$)				
138	-	-	0.02	-
139	-	-	0.29	-
141	2.12	Lost	12.58	-2.12
142	-	-	12.57	-
143	-	-	2.10	-
144	-	-	0.00	-
145	-	-	0.00	-
146	-	-	1.40	-
147	0.00	Lost	0.14	0.00
148	-	-	0.49	-
149	0.00	Lost	0.03	0.00
Retraining ($\overline{\text{NPV}} = 0.83$)				
150	0.06	Salvaged	0.16	0.10
151	-	-	12.40	-
152	-	-	0.02	-
153	-	-	30.19	-
154	-	-	0.01	-
155	2.18	Lost	37.88	-2.18
156	1.93	Lost	12.70	-1.93
157	-	-	0.02	-
158	-	-	1.61	-
159	-	-	0.51	-
Retraining ($\overline{\text{NPV}} = 0.77$)				
160	0.01	Salvaged	1.50	1.49
161	-	-	0.80	-
162	0.90	Lost	14.26	-0.90
163	-	-	13.38	-
164	-	-	1.37	-
165	6.67	Lost	30.93	-6.67
166	2.11	Lost	14.56	-2.11

167	-	-	2.96	-
168	-	-	0.01	-
169	-	-	0.04	-
170	-	-	2.46	-
171	-	-	0.03	-
172	-	-	0.97	-
Retraining ($\overline{\text{NPV}} = 0.76$)				
173	-	-	1.91	-
174	9.77	Lost (unacceptable risk)	14.17	-9.77
175	7.62	Lost	13.59	-7.62
176	0.00	Lost	14.58	0.00
177	0.17	Salvaged	12.30	12.13
178	-	-	0.03	-
179	-	-	2.71	-
180	-	-	0.05	-
181	-	-	0.01	-
182	0.00	Lost	0.07	0.00
183	-	-	0.03	-
184	-	-	0.86	-
185	10.74	Lost (unacceptable risk)	15.22	-10.74
186	-	-	1.31	-
187	-	-	12.27	-
Retraining ($\overline{\text{NPV}} = 0.81$)				
188	-	-	6.10	-
189	-	-	14.67	-
190	2.01	Lost	12.64	-2.01
191	-	-	0.04	-
192	1.57	Lost	12.74	-1.57
193	1.99	Lost	26.16	-1.99
194	2.46	Lost	13.70	-2.46
195	-	-	0.28	-
196	2.68	Lost	13.43	-2.68
197	2.31	Lost	17.20	-2.31
198	-	-	1.56	-
199	6.82	Lost	13.49	-6.82
200	2.32	Lost	15.29	-2.32
201	-	-	1.41	-
202	-	-	1.42	-
Retraining ($\overline{\text{NPV}} = 0.74$)				
203	-	-	1.44	-
204	-	-	1.43	-
205	-	-	0.02	-
206	0.04	Salvaged	1.40	1.35
207	-	-	1.40	-
208	-	-	0.05	-

209	-	-	1.44	-
210	-	-	1.45	-
211	-	-	0.06	-
212	-	-	0.01	-
213	-	-	1.47	-
214	-	-	15.30	-
215	-	-	0.34	-
216	-	-	1.54	-
Retraining ($\overline{\text{NPV}} = 0.74$)				
217	-	-	0.01	-
218	-	-	1.50	-
219	-	-	0.05	-
220	-	-	1.50	-
221	-	-	1.66	-
222	-	-	1.41	-
223	-	-	0.01	-
224	-	-	1.48	-
225	-	-	1.42	-
226	-	-	1.45	-
227	-	-	1.42	-
228	-	-	1.46	-
229	-	-	0.26	-
230	-	-	0	-
231	-	-	0.01	-
Retraining ($\overline{\text{NPV}} = 0.74$)				
232	-	-	2.80	-
233	-	-	0.01	-
234	-	-	2.19	-
235	6.86	Lost	37.76	-

D Tutorials

D.1 Understanding operational disturbance and noise

A generic process recipe consists of five tasks: ‘start’, ‘2’,’3’,’4’, and ‘end’. The process is under the influence of operational uncertainty and is controlled by GMP, which determines which tasks must be repeated in case of failure—e.g., only the failed task must be repeated until success. If a batch is performed without failure the exact recipe sequence is observed in the raw data (see A, in **Figure D.1**); however, if the batch is faulty, the *operational disturbance* (red, **Figure D.1**) is observed in the raw data (see B, in **Figure D.1**) and the recipe is

truncated—e.g., three data points are repeated inside the recipe sequence. Differently from the *operational disturbance*, the *noise* (grey, **Figure D.1**) appears between batches; all data points between batches are classified as noise. Two types of noise can be observed: One is identified as the recording of a task which belongs to the recipe but does not follow the any sequence (see C, in **Figure D.1**); the other is the recording of a task ID which does not belong to the process recipe (see D, in **Figure D.1**).

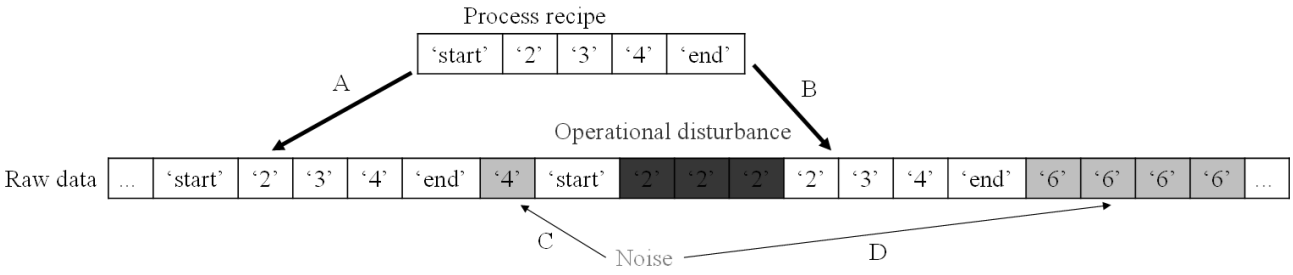


Figure D.1 Generic graphical representation showing the difference between operational disturbance (red) and noise (gray)

D.2 Data sequencing

D.2.1 Find ETS (step 3.1)

All the ETS sequences are located as following the below algorithm; the number of total ETS is unknown.

Let the vectors $sETS = [s_1, s_2, s_3, s_4]^T$ and $eETS = [e_1, e_2, e_3, e_4]^T$ be the starting and ending ETS.

- for $\forall n < N - s + 1$ calculated the distance $D^{\text{sETS}}_{n,n+s-1}$ between $sETS$ and $[\boldsymbol{D_1}(n), \dots, \boldsymbol{D_1}(n+s-1)]$
- for $\forall n < N - s + 1$ calculated the distance $D^{\text{eETS}}_{n,n+s-1}$ between $eETS$ and $[\boldsymbol{D_1}(n), \dots, \boldsymbol{D_1}(n+s-1)]$
- The position $p^{s\text{ETS}_j}$ of the sETS j in equal to p_n $\forall n$ where $D^{\text{sETS}}_{n,n+s-1} = 0$
- The position $p^{e\text{ETS}_j}$ of the eETS j in equal to p_n $\forall n$ where $D^{\text{eETS}}_{n,n+s-1} = 0$

The ETS positions set of the $p^{\text{ETS}} = \{p^{s\text{ETS}}, p^{e\text{ETS}}\}$ is saved and is used as placeholder in the sequencing.

D.2.2 Calculate primer size s_n^p (step 3.2)

The primer size at each data point n is calculated; during the sequencing the primer size adapts depending on its distance from the nearest ETS. The primer size is calculated as follows:

- $\forall n < N-1$ calculate the distance of the data point from the nearest ETS as $d_n^{\mathrm{ETS}} = \min_j |p_n - p^{\mathrm{ETS}_j}|$
- $\forall n < N-1$ use **Figure** 3.4 to determine how to calculate the primer size s_n^p

D.2.3 Calculate Wagner-Fischer distance and heterogeneity coefficient

First, the WF distance between each primer and fragments of $\boldsymbol{D_1}$ are calculated

Let P and R be the primer and recipe string vectors

- $\forall n < N-1$ and $\forall k < K - s_n^p + 1$ define the string vector $P_{k,n} = [R(k), \dots, R(k + s_n^p - 1)]$
- $\forall n < N-1$ and $\forall k < K - s_n^p + 1$ calculate the distance $D_{n,k}^{WF}$ between the $P_{k,n}$ and $[\boldsymbol{D_1}(n), \dots, \boldsymbol{D_1}(n + s_n^p - 1)]$
- $\forall n < N-1$ calculate the minimum distance $D_n^{WF} = min_k\,(D_{k,n}^{WF})$
- $\forall n < N-1$ calculate the heterogeneity coefficient $H_n = \frac{D_n^{WF}}{s_n^p}$

D.3 IDEF0

The IDEF0 notation is used to describe the interconnection between actions (*activities*) and information; as it is shown in **Figure D.2** an *activity* is defined through its description—e.g., "do X", "analyze Y"—and is identified by an ID. An *activity*—e.g., transform raw data—takes an *input*—e.g., raw data—and delivers an *output*—e.g., structured data—; the *activity* is performed following a *mechanism* and is controlled by a *control*, an example of which is "preserve data integrity". Activities can be layered by introducing sub-activities, which use all inputs, control, and mechanisms of the main activity. The identifier starts from the outer layer with A0, also called zero layer, and continues with the notation A*IJ*... in the inner layers—e.g., A41 stays for the first sub-activity of the fourth activity in the first layer and is located in the second inner layer. An exhaustive and extended description and definition of the IDEF0 method is found at http://www.idef.com/idefo-function_modeling_method/.[177]

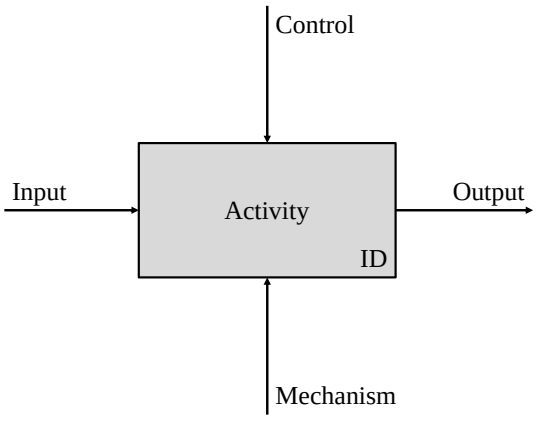


Figure D.2 Generic IDEF0 notation