Special Issue

Simulation

Simulation: Transactions of the Society for Modeling and Simulation International
1–26

DOI: 10.1177/00375497241302866
journals.sagepub.com/home/sim
Sage SCS

Data-driven reliability assessment of manufacturing systems using process mining

Jonas Friederich[1] and Sanja Lazarova-Molnar[1,2]

Abstract
Reliability analysis has long been used to understand and predict system behaviors in various industries, including manufacturing, aerospace, and energy. However, the increasing complexity and dynamics of modern systems can quickly outpace manually developed, expert-based models. Conversely, the increasing availability of data from industrial Internet of Things (iIoT) sensors and advanced control systems enables a more data-driven approach to reliability modeling, coping with the aforementioned issues. In this paper, we introduce a framework for data-driven reliability assessment of manufacturing systems using process mining. With our framework, we aim to provide a systematic approach to extract, simulate, validate, and exploit reliability models to support decisions within manufacturing systems. We demonstrate our framework using two case studies based on a flow line commonly found in today's shop floors.

Keywords
Reliability assessment, process mining, manufacturing, stochastic Petri nets, modeling & simulation

1. Introduction

In the advent of the fourth industrial revolution, often referred to as Industry 4.0 (I4.0), manufacturers are moving to a new level of value chain organization and control.[1] Physical and computational resources are seamlessly integrated to augment production efficiency, productivity, and flexibility.[2] In addition, new technologies such as the industrial Internet of Things (iIoT), Cloud Computing, Big Data, and Artificial Intelligence enable manufacturers' new ways of production planning and control. However, on the flip side of these developments is the increasing complexity of manufacturing systems. As a consequence, reliability emerges as a critical concern.[3,4] Ensuring reliable operations of manufacturing systems is crucial to avoid production disruptions, expensive downtime, and potential safety issues.

Reliability assessment is a systematic process that evaluates reliability of a product, system, or service under specified conditions, determining its ability to function without failure over a defined period of time.[5,6] Such failures not only impact performance of manufacturing systems but can also lead to accidents that compromise the safety of these systems.[7] Traditional reliability assessment employs various statistical and probabilistic methods to understand and quantify the reliability, availability, and maintainability of a system. Ideally, reliability assessment should be performed during the design phase and repeated continuously throughout the system's lifespan, especially when changes are made.[4]

A powerful approach used for reliability assessment of manufacturing systems is Modeling and Simulation (M&S). M&S provides a systematic way to represent, analyze, and predict system behaviors without imposing high costs or operational disruptions. M&S can be used to simulate different operational scenarios, failure modes, maintenance strategies, and process variations, helping stakeholders understand their potential impact on system reliability.[8,9] It enables the testing of the system under varying conditions to identify potential bottlenecks and other weaknesses.

Traditional reliability assessment, which employs M&S and other methods, heavily depends on expert knowledge of the system under study.[10] This dependency results in several limitations. For example, experts are required to

[1]Maersk Mc-Kinney Moller Institute, University of Southern Denmark, Denmark
[2]Institute of Applied Informatics and Formal Description Methods, Karlsruhe Institute of Technology, Germany

Corresponding author:
Jonas Friederich, Maersk Mc-Kinney Moller Institute, University of Southern Denmark, Campusvej 55, Odense 5230, Denmark.
Email: jofr@mmmi.sdu.dk

analyze and model system topology, failure and repair distributions, and other factors. However, due to the dynamic and complex nature of modern manufacturing systems, expert knowledge, while invaluable, can become a bottleneck.[11] Another shortcoming of traditional reliability assessment is the static nature of manually developed models, which fail to account for changes in a system over time. As real systems evolve, manually developed models can quickly become outdated and require updating. This manual model-updating process is labor-intensive and tedious, particularly in systems where frequent changes in system topology and configuration occur.[12]

The advancing automation of manufacturing processes and the deployment of iIoT sensors and actuators enable manufacturers to control, monitor, and analyze production lines in real time. We can use data collected in these production lines to make reliability assessment more dynamic and data-driven. Lazarova-Molnar et al.[10] note that data-driven techniques, when effectively harnessed, have the potential to significantly improve the reliability of complex systems. With data-driven reliability assessment, we can address the challenges of traditional reliability assessment mentioned above by automatically extracting reliability models and continuously updating them whenever changes in the real system occur.

Process mining (PM) is a potential supporting technique that can be used for data-driven reliability assessment. PM enables production processes to be reconstructed and analyzed based on digital traces in information systems. PM, thus, facilitates to model the implicit and otherwise hidden process knowledge, contained in data, to make it tangible and transportable.[13] Since PM is a data-driven technique, less expert knowledge of the system under study is required, and the extracted models are updated as the system configuration or topology changes.

While PM has been applied in previous works for automated generation of discrete-event simulation models of production systems,[14,15] its application for reliability assessment of manufacturing systems remains unexplored. To the best of our knowledge, this work represents the first effort to combine and advance both reliability engineering and PM, making reliability assessment of manufacturing systems more dynamic and data-driven.

In this article, we introduce a framework for data-driven reliability assessment of manufacturing systems using PM. Our framework takes event and state logs collected within a manufacturing system as input and automatically extracts a reliability model. We use stochastic Petri nets (SPNs) as modeling formalism for their ability to represent complex, discrete processes, subject to uncertainty and randomness.[16] We simulate extracted reliability models using discrete-event simulation (DES) and validate them using streaming data from the real system. We use the extracted models to support various decisions regarding, e.g., purchase decisions and maintenance scheduling.

The remainder of this article is structured as follows. In section 2, we provide relevant background to reliability assessment of manufacturing systems, reliability-relevant data collection in such systems, and PM as a data-driven technique to enable automatic reliability model extraction. Following, in section 3, we discuss related work. In section 4, we present our proposed framework for data-driven reliability assessment of manufacturing systems. In section 5, we provide two case studies demonstrating the application of our proposed framework. Finally, in section 6, we summarize and conclude our work, and in section 7, we provide an outlook and discuss potential future work.

2. Background

2.1. Reliability assessment of manufacturing systems

The adoption of new technologies and paradigms increases the complexity of manufacturing systems, making it more difficult to maintain the systems and to identify possible vulnerabilities that affect their reliability. To this end, reliability assessment includes a number of techniques for planning and monitoring reliable manufacturing systems and for detecting such vulnerabilities.[4] The goal of reliability assessment of any system operated in industry is to evaluate the system's reliability and to ensure a continuous operation without failures. Thus, overall production costs and downtime can be kept to a minimum if reliability is maintained at a high level.[7]

Reliability assessment is a systematic approach to identify and evaluate the causes and frequency of failures and to reduce or control the effects of failures to ensure good system performance.[5] Failures not only affect the performance of a manufacturing system but can also cause accidents.[7] Traditional reliability assessment evaluates the frequency of such events during the design phase of a system. To maximize profit and safety, reliability assessment of such a system must be performed during the design phase and applied until the system is finally replaced. Ideally, a new assessment should be performed whenever changes are made to the system.

In this article, we consider manufacturing systems. The term *manufacturing* is used to denote a general activity to transform raw material into consumable products such as cars or smartphones. *System* refers to a set of resources involved in this transformation process and their dependencies which allows all resources to work together.[17] These resources can be grouped into categories, such as hardware (e.g., machines and tools), humans (e.g., operators, maintenance engineers, and supervisors), and software (e.g., Manufacturing Execution System (MES), Enterprise Resource Planning (ERP), Programmable Logic Controller (PLC), and Supervisory Control and Data Acquisition (SCADA)). All three resource types pose unique challenges to their reliability assessment, which

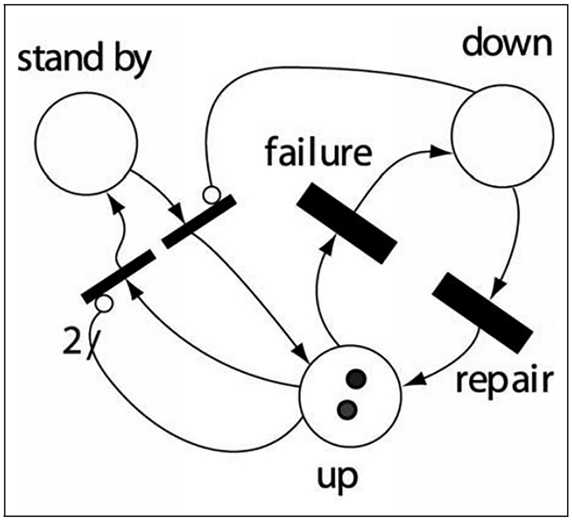

Figure 1. Sample Petri net for reliability modeling of two repairable resources connected in series.[19]

has led to a variety of methods to address these challenges. Reliability assessment can be carried out using both qualitative and quantitative methods, which involve reliability modeling and reliability analysis, respectively.

Qualitative reliability modeling involves the use of models, diagrams, or other visual representations to analyze and understand the reliability of a manufacturing system and its resources.[18] Qualitative models are often used to identify potential failure modes and assess the impact of these failures on system performance. These models are typically based on the experience of experts in the field and may involve the use of fault trees, Petri nets, or other similar techniques to evaluate the likelihood of different failure scenarios. Figure 1 shows a sample Petri net of a simple system that contains two repairable resources connected in series.

Quantitative reliability analysis involves the use of mathematical and statistical tools to evaluate the reliability of a system and its resources.[18] This analysis typically involves the use of data on system performance, such as failure rates, mean time between failures, and other relevant metrics, to quantify the likelihood of failure over time. Quantitative analysis can help identify potential weaknesses in a system and can be used to design more reliable systems that can better withstand the stresses of the manufacturing environment.

One key advantage of qualitative modeling is its ability to identify potential failure modes and assess the impact of these failures on system performance. Quantitative reliability analysis, on the contrary, provides a more rigorous and precise assessment of system reliability, based on empirical data and statistical analysis. In practice, both qualitative and quantitative methods are often used in combination to provide a comprehensive reliability assessment of manufacturing systems.

2.2. Reliability-relevant data collection in manufacturing systems

Nowadays, production facilities generate a wide range of data, and understanding the types of data generated is essential for data-driven reliability assessment of manufacturing systems. Zschech[20] proposes a comprehensive taxonomy that addresses recurring data analysis problems in production analytics, providing valuable insights into relevant data types. The taxonomy distinguishes between condition monitoring data, event data, metadata, and business data for hardware resources within a production system. In Table 1, we extend the taxonomy by providing example attributes for each of the three resources types—hardware, human, and software—within a manufacturing system, with a specific focus on reliability.

Condition monitoring data involve measurements specific to production resources. Sensors are commonly employed to collect data for physical and human resources, while software resources can generate monitoring data programmatically. Event data capture different occurrences on a resource and the corresponding actions taken.[20,21] Event

Table 1. Data types and example attributes for data collected within manufacturing systems.

Data type	Hardware resource	Human resource	Software resource
Condition monitoring data	Pressure, vibration, temperature, humidity, power consumption	Biometric data (heart rate, body temperature), fatigue level, stress level, workload, ergonomic data	Response time, application instances, request rate, CPU usage, garbage collection
Event data	Faults, failures, errors, operations, repairs, configurations, maintenance	Operations, failures, trainings, breaks	Faults, failures, errors, operations, updates, configurations, maintenance
Metadata	Machine type, location, machine manufacturer, installation date, maintenance schedule	Position/role, experience, education, department, hire date	Software type, developer, maintenance schedule
Business data	Quality measures, performance indicators, processes	Quality measures, performance indicators, processes	Quality measures, performance indicators, processes

data encompass events such as faults, failures, errors, operations, updates, and configurations. Lazarova-Molnar et al.[3] note that the meaning of failure as a complete system standstill is similar for all three resource types. However, the interpretation of errors and faults may slightly differ between the three domains. Metadata refers to contextual details about the specific resources of manufacturing systems, while business data offer broader contextual information about the manufacturing system itself. It includes aggregated quality measures, performance indicators, and information about business processes.

To collect the aforementioned data, researchers and practitioners can leverage multiple data sources, including:

- Intrinsic data sources: Manufacturing systems generate intrinsic data through built-in sensors, actuators, and control systems such as SCADA and PLC. These data can be readily accessed and collected from the manufacturing system infrastructure itself. Examples include sensor data from production machines, control system logs, and real-time process data.
- Company information system: The company information system encompasses data acquired from information systems, databases, or platforms. This may include data from ERP systems, MESs, Supply Chain Management (SCM) systems, or other sources.
- Manual data collection: In some cases, certain data may not be automatically captured by manufacturing system infrastructure. In such situations, manual data collection methods, such as surveys, interviews, and checklists, can be employed to gather specific information. For instance, data related to operator observations, subjective assessments, or system configuration details may require manual collection.

In this article, we utilize intrinsic data sources and data from the company information system to extract reliability models capturing the production process and the involved hardware resources. We distinguish between state logs and event logs, both having a time series format. Each data record includes a timestamp and type-specific information. The availability of more data enhances the potential level of detail in the extracted reliability models.

State logs provide a record of operational state changes in the individual resources of a system, as well as the system itself. These data can include information such as up-times, down-times, working, idle, failed, and repaired states. While state logs provide a low-level source of information, they lack explanatory details regarding system behavior and failure causes. Nevertheless, collecting this type of data requires less effort due to its reduced complexity.

Event logs comprise discrete events generated by the resources and the system. These events mark the initiation and completion of activities relevant to a specific reliability assessment study, such as raw material preparation, material transport, or operations at assembly cells. Each event record should include a case identifier, enabling grouping of events belonging to the same product instance's production trace. Event logs offer valuable insights into system behavior, facilitating the creation of realistic and accurate reliability models. However, acquiring detailed event logs requires extracting information from resources, potentially involving interfaces with diverse manufacturers. In addition, in cases where resources cannot provide event data through an interface, the installation of sensors becomes necessary.

We provide a mathematical formalization of state logs and event logs later in this article in Section 4.

2.3. PM for data-driven model extraction

PM is an emerging interdisciplinary field that lies at the intersection of data science, business process management, and information systems. It aims to extract valuable insights and knowledge from event data generated during the execution of processes within an organization. By analyzing and visualizing these event logs, PM enables researchers and practitioners to understand, improve, and optimize business processes, leading to enhanced efficiency, reduced operational costs, and better decision-making.[13]

PM can be divided into three main disciplines: process discovery, conformance checking, and process enhancement:

- Process discovery involves the automated extraction of process models or maps from event logs.[13] Various process discovery algorithms have been developed to create process models that represent the actual behavior of processes, allowing organizations to gain insights into the underlying workflows and identify bottlenecks or deviations from expected behavior.
- Conformance checking aims to compare the observed behavior of a process with the modeled or expected behavior.[13] This enables organizations to assess how well their processes are being executed and identify any deviations or non-compliance issues. Conformance-checking techniques help identify root causes of inefficiencies, compliance violations, and other process-related problems, leading to targeted process improvements.
- Process enhancement involves identifying areas for optimization, re-engineering inefficient process steps, and making data-driven decisions to streamline operations.[13] It also supports the adoption of

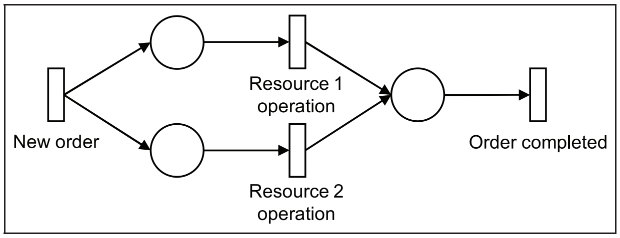

Figure 2. Sample Petri net extracted using process discovery.

continuous improvement methodologies such as *Lean* and *Six Sigma*, as PM provides a factual basis for measuring and managing process changes over time.

Figure 2 shows a sample Petri net capturing a production process involving two production resources operating in parallel. The Petri net was extracted based on a event log using a process discovery algorithm. In this article, we utilize PM, more specifically a combination of process discovery and process enhancement, to support the automated extraction of reliability models for manufacturing systems from data.

3. Related work on data-driven M&S of manufacturing systems

In this section, we discuss the evolution of data-driven simulation modeling in manufacturing systems, emphasizing advancements in both system modeling and reliability assessment.

Data-driven modeling leverages real-world data to construct, calibrate, and validate models, especially in the complex and dynamic domain of manufacturing.

For example, Wang et al.[22] propose a data-driven simulation modeling methodology that automates modeling of production systems and facilitates rapid adjustments in response to dynamic requirements and real-time data. Zou et al.[23] introduce a novel data-driven stochastic modeling method designed to identify energy-saving opportunities within manufacturing systems and evaluate their impact on production. Resman et al.[24] present a comprehensive five-step approach to planning and modeling data-driven digital twins of manufacturing systems and their processes. All of these contributions rely on data for calibrating and validating simulation models, as well as for assessing sensitivity of model parameters. However, these methods assume an initial manually derived model created by a domain expert.

Recent developments in the field aim to go beyond calibration and validation by automatically extracting the initial and explicit model structure from data. For instance, Milde and Reinhart[25] develop an approach that simultaneously extracts material flow information, estimates model parameters, and identifies control policies from event logs of manufacturing systems. Similarly, Lugaresi and Matta[15] propose an approach for the automated extraction of simulation models for manufacturing systems based on event logs. Finally, in the work by Friederich et al.,[26] we propose a novel approach to generate data-driven reliability models of complex manufacturing systems from event and state logs.

PM, as an inherently data-driven technique, proves to be highly suitable for extracting explicit models for simulation purposes. Ferronato et al.[14] propose a framework for automated generation of DESs based on Petri nets extracted from event logs using PM. Furthermore, in the work by Friederich et al.,[27] we define data requirements and outline an approach for extracting dynamic simulation models of manufacturing systems using PM. Building upon these foundations, in the work by Friederich et al.,[28] we introduce a comprehensive framework for data-driven digital twins of manufacturing systems using PM.

In addition to data-driven modeling of manufacturing systems, some researchers have proposed data-driven approaches for reliability modeling. For example, Lazarova-Molnar et al.[18] present a method for data-driven fault tree modeling tailored for reliability assessment of cyber-physical systems. This work was extended by Niloofar and Lazarova-Molnar[29] to include expert knowledge and further expanded by Niloofar and Lazarova-Molnar[30] to incorporate data from multiple sources sharing similar properties. In the work by Friederich and Lazarova-Molnar,[12] we propose a framework for data-driven reliability assessment using event, state, and condition monitoring data.

This article builds upon our previous conference paper,[31] where we introduce a PM-based methodology to extract accurate reliability models from event and state logs. In this article, we not only extract reliability models from data but also validate them and demonstrate how these models can be applied to support decision-making processes in a manufacturing context.

4. Framework for data-driven reliability assessment

In this section, we present our framework for data-driven reliability assessment of manufacturing systems. The aim of this framework is to enable automated extraction of reliability models using data collected from manufacturing systems and to use these models for decision-making.

Figure 3 illustrates the general process our framework follows. Starting with the system of interest, reliability-relevant data are collected. These data then undergo preprocessing to ensure its suitability for data-driven reliability modeling. During preprocessing, the data are transformed into a format suitable for model extraction methods. The preprocessed data are then utilized to extract reliability models. Once derived, these models are

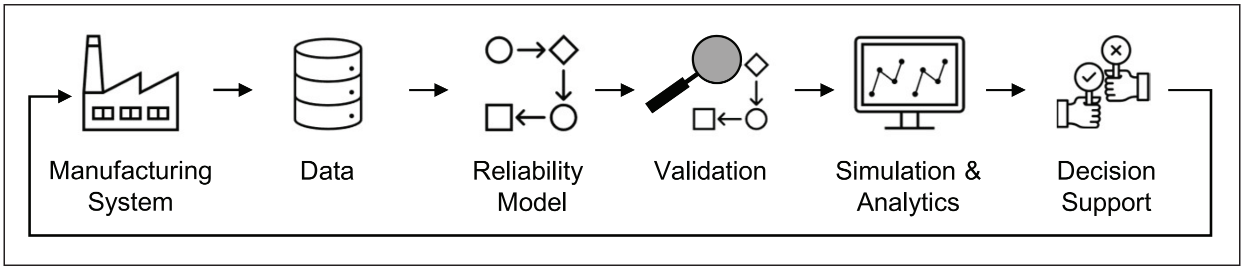

Figure 3. Data-driven reliability assessment process.

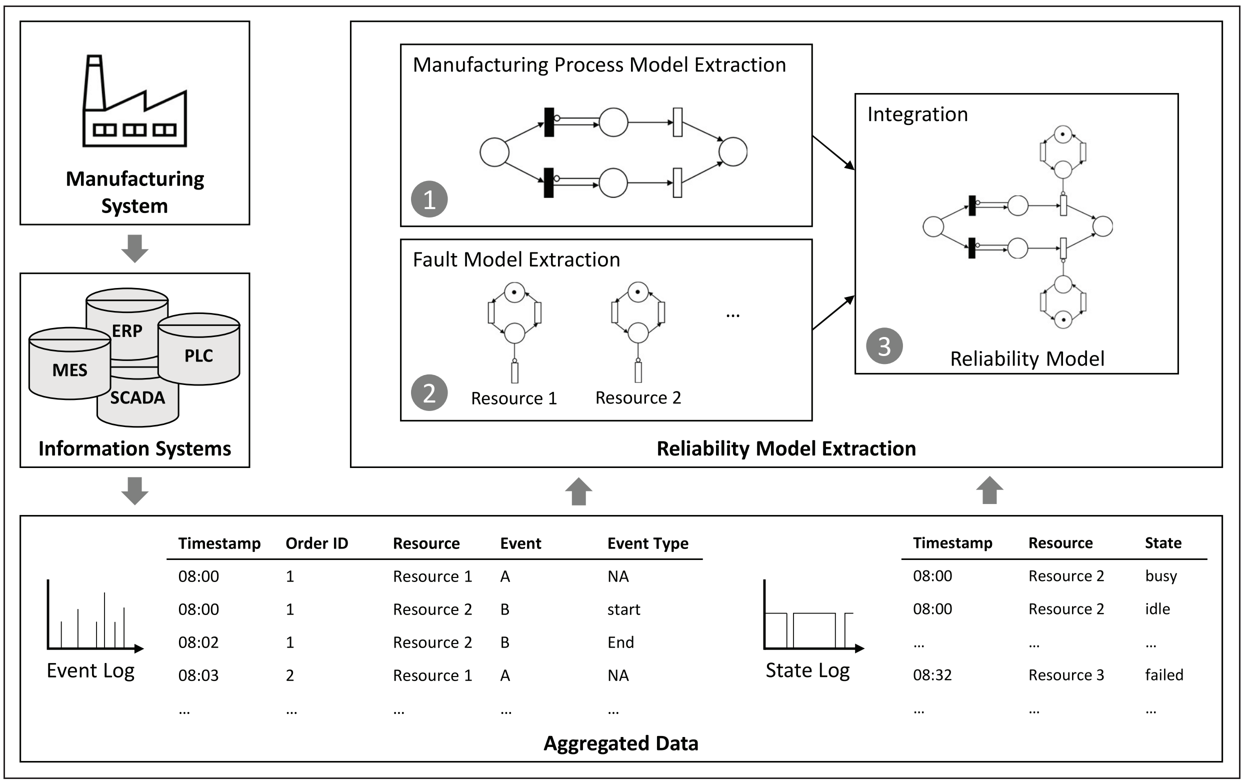

Figure 4. Overview of the model extraction process.

validated to ensure their accurate representation of the real system. Once validated, simulation and data analytics techniques are employed to run experiments aiming to support decision-making processes. These decisions may include system configuration, purchase decisions, and maintenance scheduling.

In the following, we begin by presenting our proposed approach for extracting a reliability model of a manufacturing system using PM. Subsequently, we briefly outline how we simulate extracted models. Furthermore, we discuss the validation of the extracted models and propose a systematic approach to ensuring the continuous validity of models deployed in production. Finally, we demonstrate how these extracted and validated models can be used to inform various decision-making processes.

4.1. Model extraction

In Figure 4, we illustrate the model extraction process. Starting from a manufacturing system, the system's data are collected and distributed to the company's information systems. The data captured by such systems are then aggregated and synthesized in event logs and state logs. Then, these data are used to extract a reliability model.

A manufacturing system event log is a valuable source of data that captures information from the various processes involved in a manufacturing system. Typically, event logs are obtained from MESs or ERP systems. In the context of manufacturing, we adopt the general assumptions about processes as stated by van der Aalst,[13] with slight adjustments to align with the manufacturing domain. The following assumptions describe the characteristics of a manufacturing process:

- A manufacturing process consists of production *orders*.
- An *order* is characterized by a sequence of *events*.
- Events can be of two *types*: Some events denote the initiation and completion of production activities, which have a duration associated with them. Other events capture logical decisions that are instantaneous and do not have a specific duration.
- Each event has a corresponding *timestamp* and *resource* that is seized and released.

Considering the abovementioned assumptions, we define an event log EL as a set of event entries in the following way:

$$EL = \{E_0, E_i, ..., E_m\},\ i = 1, ..., m.$$

Each event log entry is defined as a tuple $E_i = (t, o, r, e, \tau)$, where:

- t is the timestamp of the event, indicating the exact time at which it occurred.
- o is the order identifier, a unique numerical value assigned to each production order, enabling individual order tracking.
- r is the resource identifier, a unique string used to identify the specific production resource involved in executing the event. This could include entities such as Automated Guided Vechicles (AGVs) or assembly cells.
- e is the event identifier, a unique string that indicates the initiation or completion of an activity, such as transport or assembly operations, or denotes logical decisions such as routing choices for a production order.
- τ represents the event type, where $\tau \in \{start, end, NA\}$; *start* when the event marks the initiation of an activity, *end* when it indicates the completion of an activity, or *NA* when the event describes a logical, instantaneous decision.

A resource state log captures state changes in the resources within a manufacturing system. Data for such logs can be acquired by PLC or SCADA systems. For our approach, we define a state log SL as a set of state change entries as follows:

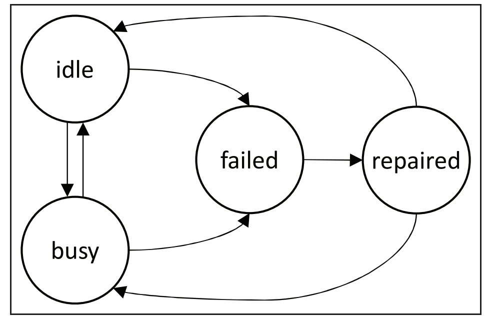

Figure 5. Possible operational state changes.

$$SL = \{S_0, S_i, ..., S_n\},\ i = 1, ..., n$$

Each state log entry is defined as a tuple $S_i = (t, r, s)$, where:

- t is the timestamp indicating the time at which the state change occurred.
- r is the identifier for the resource that changes its state.
- s is the new state that the resources transitioned to, where $s \in \{busy, idle, failed, repaired\}$; *busy* when the resource is actively engaged in an production activity, *idle* when the resource is available but not currently in use, *failed* when the resource is inoperable, and *repaired* when the resource has been repaired and restored to an operational state.

Figure 5 displays possible operational state changes in resources, as captured by the state log.

The manufacturing process model is extracted from an event log. Subsequently, the manufacturing process model is enriched with resource fault models that capture probability distributions describing failures and repairs of resources, estimated from the resource state log. We use SPNs as modeling formalism for our approach, due to their ability to handle complex manufacturing processes subject to uncertainty and randomness. Formally, the class of SPNs that is considered in this article can be defined as follows:

$$SPN = (P, T, A, m_0)$$

where:

- $P = \{P_1, P_2, .., P_p\}$ is the set of places, drawn as circles.
- $T = \{T_1, T_2, .., T_q\}$ is the set of transitions along with their distribution functions or weights, drawn as bars.
- $A = A^I \cup A^O \cup A^H$ is the set of arcs, where A^O is the set of output arcs, A^I is the set of input arcs, and A^H

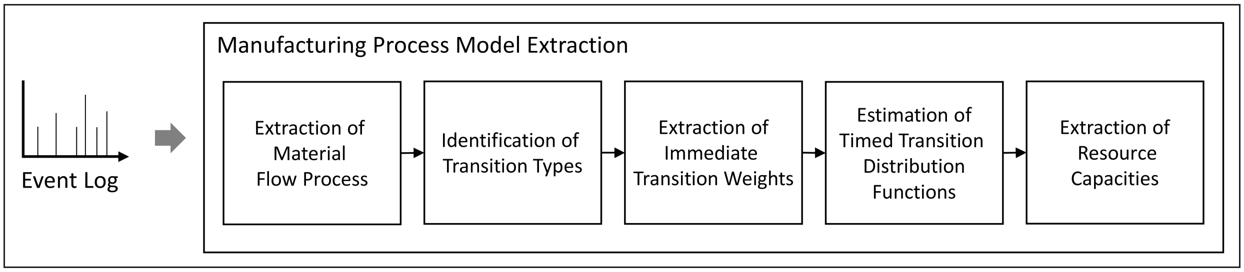

Figure 6. Extraction of the manufacturing process model.

is the set of inhibitor arcs, and each of the arcs has a multiplicity assigned to it.
- m_0 is the initial marking of the Petri net.

Each transition is defined as $T_i = (e, r, n, f, type)$ and corresponds to an event $\{E.e\}$, $\forall E \in EL$ in the event log. Here, the name of the event e serves as the label for the transition T_i. r is the resource, n is the frequency (i.e., the number of times the transition fired), and f is a probability distribution function if the corresponding transition is timed and a firing weight if it is immediate. $type$ is the type of the transition where $type \in \{timed, immediate\}$.

The set of arcs are defined such that:

$$A^O = \{a_1^o, a_2^o, ..., a_k^o\},\ A^I = \{a_1^i, a_2^i, ..., a_j^i\}\ and$$

$$A^H = \{a_1^h, a_2^h, ..., a_i^h\}$$

where

$$A^H, A^O \subseteq P \times \left(T \cup^I\right) \times \mathbb{N},\ A^I \subseteq \left(T \cup^I\right) \times P \times \mathbb{N}.$$

In this context, for a transition T_i to be enabled to fire, it must satisfy the following two conditions: (1) for each input place P_i of transition T_i, the number of tokens in P_i must be greater than or equal to the multiplicity of the corresponding input arc for T_i and (2) the number of tokens in each place P_i connected via inhibitor arcs to T_i is less than the multiplicity of the corresponding inhibitor arc. Upon firing, transition T_i removes tokens from its input places and creates tokens in its output places, whose numbers are according to the corresponding output arcs' multiplicities. Immediate transitions fire instantaneously when enabled. In case of competition, the firing weights f of immediate transitions are used to determine which transition fires. The firing time of a timed transition is determined by its probability distribution function f. We set the memory policy for all timed transitions to be *race age*, meaning the elapsed enabling time is remembered when a timed transition becomes disabled before it fires.[32]

4.2. Extraction of the manufacturing process model

We extract the manufacturing process model in five steps, with each step increasing the model's level of detail (Figure 6). The material flow process represents the path that production orders follow through the system. From a SPN modeling perspective, such a process might consist of several timed and immediate transitions. Timed transitions may correspond to activities such as transport of material or the operation of an assembly cell. Immediate transitions, on the contrary, represent logical decisions such as routing decisions in scenarios involving concurrent operations or other events without activities. Each immediate transition is associated with the frequency of the corresponding events in the event log, which enables the extraction of firing weights. The activity durations of timed transitions can be described using probability distribution functions, enabling the modeling of various time-related aspects.

The manufacturing process may also encompass resources with specific capacities, such as buffers or assembly cells. These characteristics are typically represented using inhibitor arcs, which prevent certain transitions from firing. Figure 7 provides an overview of the mentioned concepts and illustrates how we will model them utilizing data provided by a manufacturing system.

In the following, we describe each of the mentioned modeling steps in detail.

4.2.1. Extraction of the material flow process.

The material flow process represents the path that production orders follow through the manufacturing system. Based on the event log, we apply the α *-miner* algorithm[33] to extract the PN that represents the material flow process of the system. The α *-miner* is a popular process discovery algorithm capable of extracting PNs consisting of an initial marking describing the initial state of a model, the actual process model, and a final marker describing the final state of a model. However, the algorithm is not able to detect loops and to distinguish between implicit and required places which might result in additional non-required places in a discovered PN. If the system under study exhibits such characteristics, we can employ more advanced process

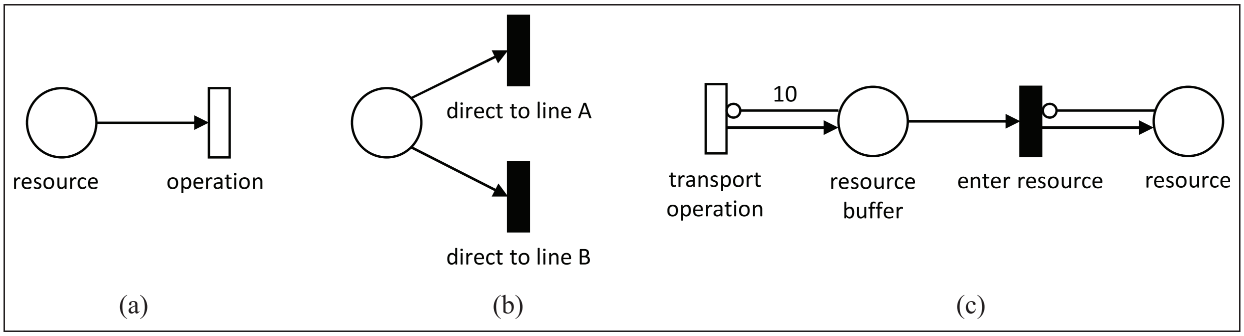

Figure 7. SPN modeling concepts for manufacturing processes: (a) Timed transition to represent an operation activity; (b) immediate transitions to represent routing decisions; and (c) use of inhibitor arcs to represent resource capacities.

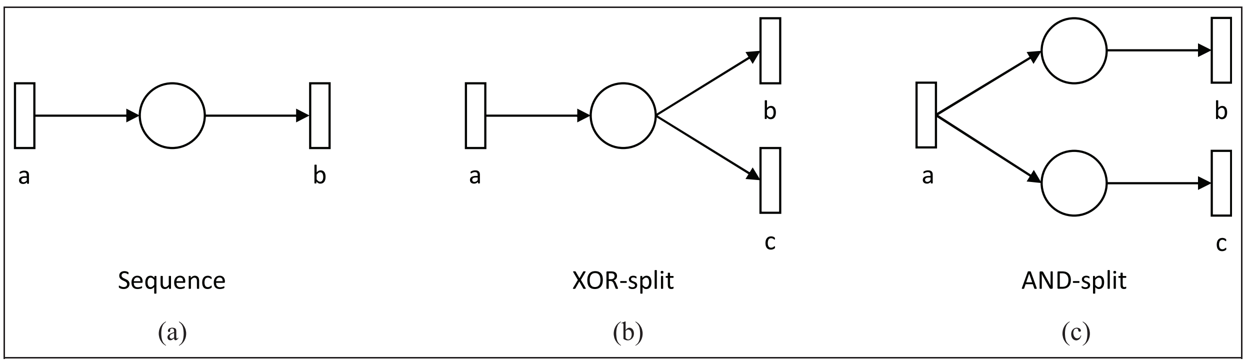

Figure 8. PN patterns that can be identified by the a-miner: (a) Sequence; (b) XOR-split; and (c) AND-split.

discovery algorithms such as the *HeuristicsMiner*[34] or the *Inductive Miner*[35] to extract the material flow process.

The α *-miner* algorithm essentially consists of two steps: (1) identify ordering relations in the log and (2) convert those relations to a PN. There are four types of ordering relations that the α-miner can detect:

- Directly follows ($a > b$): if activity a is directly followed by activity b.
- Sequence ($a \rightarrow b$): if $a > b$ and not $b > a$.
- Parallel ($a \,||\, b$): if both $a > b$ and $b > a$.
- Choice ($a \# b$): if neither $a > b$ nor $b > a$.

Figure 8 displays how the identified relations are then converted into a PN. The left PN corresponds to a sequence pattern ($a \rightarrow b$), the middle PN to a XOR-split pattern ($a > b$, $a > c$ and $b \# c$), and the right PN to an AND-split pattern ($a > b$, $a > c$ and $b || c$). A detailed description of the algorithm is provided by van der Aalst et al.[33]

4.2.2. Identification of transition types. Existing process discovery algorithms are primarily designed for extracting basic place/transition PNs and do not specifically handle the extraction of SPNs. Therefore, algorithms such as the α *-miner* extract PNs without distinguishing between timed and immediate transitions. Consequently, to incorporate timing information into the extracted material flow process model, we need to assign transitions as either timed $T_i.type = timed$ or immediate $T_i.type = immediate$.

To determine whether a transition in timed or immediate, we use the following approach: Transitions representing the arrival of new production orders can be identified by their lack of input arcs $A^I_{T_i} = \emptyset$. Such transitions are set as timed transitions. In addition, transitions representing activities performed by production resources can be identified based on their corresponding events $E_i.e$ in the event log EL. If these events have *start* or *end* as their event types $E_i.\tau$ (i.e., $E_i.\tau = startORE_i.\tau = end$), the corresponding transitions are considered timed transitions. All remaining transitions are designated as immediate transitions (Algorithm 1).

4.2.3. Extraction of immediate transition weights. Transition frequencies $T_i.n$ are the number of times an activity associated with each transition was executed based on the information available in the EL. Such frequencies provide useful quantitative information to the extracted model that can be used to assess the system's performance to calculate firing weights/probabilities $T_i.f$ of immediate transitions. We use the previously described directly follows relationships of activities in the EL (e.g.,

Algorithm 1: Extraction of transition types, immediate transition weights, and timed transition probability distributions

Input: event log *EL*, material flow Petri net *PN*
Output: *SPN* with identified transition types, transition weights, and distributions
for T_i *in SPN* **do**
  if $A^I_{T_i} = \varnothing$ **then**
    $T_i.type \leftarrow timed$
    $T_i.f \leftarrow estimateArrivalTimeDistribution(EL)$
  if $T_i.e \Leftrightarrow E_i.\tau \in start \vee E_i.\tau \in end$ **then**
    $T_i.type \leftarrow timed$
    $T_i.f \leftarrow estimateResourceActivityDistribution(EL)$
  else
    $T_i.type \leftarrow immediate$
    $T_i.n \leftarrow getDirectlyFollowsFrequency(T_i, T_{following})$
  end
end

frequency of activity a being followed by activity b) to extract transition frequencies and add them to the SPN (Algorithm 1).

4.2.4. Estimation of timed transition distribution functions. After determining the transition types, probability distribution functions of timed transitions can be estimated. Each timed transition has a probability distribution function $T_i.f$ that determines its firing times. We utilize the information in the *EL* to estimate these functions.

For the transitions representing the arrival of new production orders, we first need to extract the inter-arrival times of new orders. To do so, we calculate the time differences between the timestamps of arrival events $E_i.t$ in the *EL* that correspond to the specific transition in the PN. The calculated inter-arrival times are stored in a list. To estimate the theoretical probability distributions that extracted inter-arrival times follow, we use the Maximum Likelihood Estimation (MLE) method.[36] MLE estimates the parameters of a given probability distribution by maximizing a likelihood function, such that the assumed theoretical distribution best describes the observed data. We fit distributions commonly used in manufacturing for representing arrival times such as the exponential, normal (folded), and gamma distributions to the extracted inter-arrival times. We assess the goodness of fit by using the sum of squared errors (SSE) between the data and the fitted distributions. The probability distribution with the lowest SSE is set to be the probability distribution function of the corresponding transition $T_i.f$.

For transitions representing activities performed by production resources, we first need to extract their activity durations from the *EL*. The beginning of a resource activity is marked by the event type ‘‘start’’ ($E_i.\tau = start$), and the end of the activity is marked by the subsequent event type ‘‘end’’ ($E_i.\tau = end$) for each event $E_i.e$ in the *EL*.

Algorithm 2: Extraction of resource capacities

Input: *SPN*, *EL*
Output: *SPN* with resource capacities
$R \leftarrow \{T.r\}, \forall T \in SPN, r =$ *buffer* $\vee$ *assembly cell*; // resources with capacity
for *resource* $\in R$ **do**
  $currentLoad \leftarrow 0$
  $loadTS \leftarrow [\,]$; // time series of resource capacities
  for $E_i \in EL$ **do**
    if "*enter resource*" $\in E_i.e$ **then**
      $currentLoad \leftarrow currentLoad + 1$
      $loadTS.add(currentLoad)$
    end
    if "*exit resource*" $\in E_i.e$ **then**
      $currentLoad \leftarrow currentLoad - 1$
    end
  end
  $resourceCapacity \leftarrow max(loadTS)$
  $SPN.addResourceCapacity(resourceCapacity)$;
  // add resource capacity to corresponding inhibitor arc in SPN
End

The time difference between these two events is the duration of the resource activity. Activity durations that capture the failure of a resource are not considered (i.e., when the resource fails during operation, we discard the corresponding activity duration). The calculated activity durations for each transition are stored in a list. To estimate the theoretical probability distributions that extracted activity durations follow, we again use the MLE method using SSE to assess the goodness of fit.

Algorithm 1 shows the process of identifying transition types, adding weights to immediate transitions and the estimation of timed transition distribution functions.

4.2.5. Extraction of resource capacities. We use inhibitor arcs to model capacities of resources in the SPN (Figure 7). To determine locations of inhibitor arcs, we consider transitions where the resource associated with the transition, $T_i.r$, is, e.g., a buffer or an assembly cell as resources with capacities.

Algorithm 2 shows the calculation of resource capacities. For each entry E_i in the *EL*, *currentLoad* stores the current load of a resource as time passes. Whenever the event of an event log entry $E_i.e$ indicates that a production order is entering the resource, the current load is increased by one and added to the time series array *loadTS*. When an event indicates that a production order is exiting the resource (i.e., the production order is entering the next resource), the current load is reduced by one. The resource capacity is the maximum load over time recorded in *loadTS*. Finally, the extracted resource capacity is set as the cardinality of the corresponding inhibitor arc in the SPN.

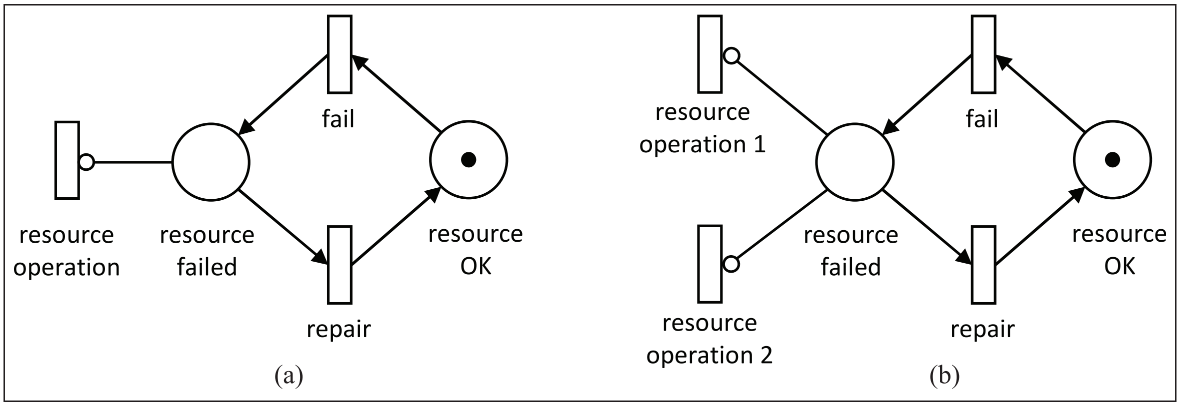

Figure 9. Exemplary fault models for a resource conducting one (a) and two (b) operations.

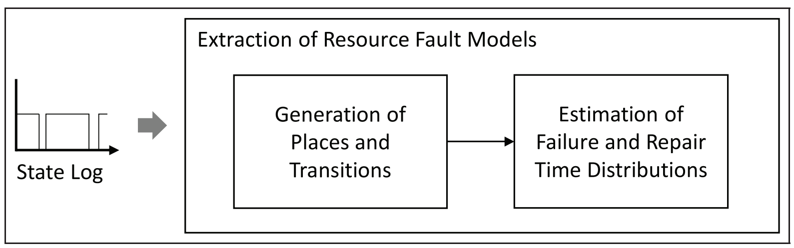

Figure 10. Extraction of resource fault models.

4.3. *Extraction of resource fault models*

Based on the state log, we extract fault models for each resource. Figure 9 shows an exemplary SPN for such a model. The initial marking in the *resource OK* place represents a fully operational resource. A timed transition represents a failure after a random amount of time sampled from a probability distribution. Once a token is created in the *resource failed* place, the resource is defect and needs to be repaired. By using an inhibitor arc to block the operation of the corresponding resource during repair, the model can be integrated into a manufacturing process model. In case the same resource is executing more than one operation, we add inhibitor arcs from the failed place to the corresponding operation transitions. The repair is represented by another timed transition which also has a probability distribution function that describes the repair duration.

We extract the fault models in two steps (Figure 10, Algorithm 3). First, the two necessary places (''resource OK'' and ''resource failed'') and transitions (''fail'' and ''repair'') are generated for each resource in the state log $\{S.r\}$, $\forall S \in SL$. The fault models are then integrated in the manufacturing process model. This is done by connecting them with transitions that utilize the corresponding resource $T_i.r$ using inhibitor arcs.

Second, similarly to the estimation of timed transition probability distribution functions of the manufacturing process model, we estimate the probability distribution functions in the fault models using the operational state changes in the SL. The failure of a resource is marked by the transition of the resource to the ''failed'' state ($S_i.s = failed$) and the repair by the subsequent transition to the ''repaired'' state ($S_i.s = repaired$). The time difference between the change from a busy ($S_i.s = busy$) or idle ($S_i.s = idle$) to the failed state is the time to failure and the time difference between the failed and the subsequent repaired state is the time to repair.

Algorithm 3: Extraction of resource fault models

Input: *SPN*, *SL*
Output: *SPN* with fault models
  $R \leftarrow \{S.r\}, \forall S \in SL$
for *resource* $\in R$ **do**
  faultModel $\leftarrow$ *generateFaultModels*(*resource*)
  estimateFailureDistributions(*faultModel*,*resource*,*SL*)
  estimateRepairDistributions(*faultModel*,*resource*,*SL*)
  integrateToSPN(*SPN*,*faultModel*)
End

To estimate the theoretical probability distributions of the extracted time to failures and time to repairs, we again use the MLE method. We fit distributions such as normal (folded), lognormal, Weibull, and exponential. We evaluate the goodness of fit using SSE between the empirical data and the fitted probability distributions.

4.4. *Model simulation*

Extracted reliability models are simulated using DES. DES is a popular method for simulation of SPNs due to its

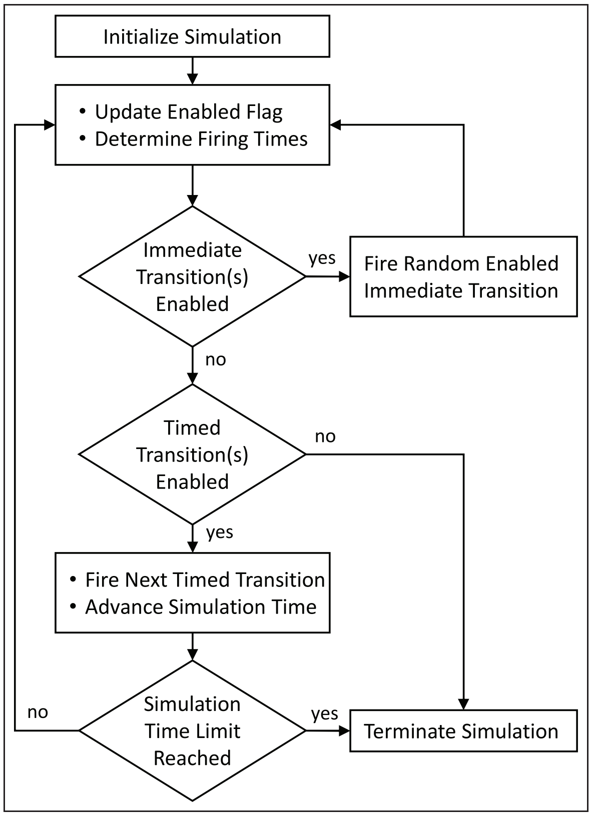

Figure 11. SPN simulation procedure flowchart.

versatility and efficiency in simulating complex systems. In DES, the simulation time is divided into discrete time intervals, and events are triggered based on the state of the system and the occurrence of random variables. This approach allows for the modeling of stochastic systems with a high degree of realism, as it can capture the effects of randomness and variability in the system. DES is particularly useful in simulating SPNs, as it can effectively handle the probabilistic transitions and random variables that are inherent in these network models.[37]

Figure 11 provides a high-level flowchart representation of the procedure used in this article for DES of SPNs. The simulation process is initialized by setting the simulation time to 0 and defining the maximum time for the simulation run. Next, transitions are evaluated to determine whether they are enabled or disabled based on the current marking of the SPN. Once transitions are evaluated, firing times are determined for all enabled transitions. If any immediate transitions are enabled, they are fired immediately without any time delay. For timed transitions, the simulation fires the transition with the next firing time and advances the simulation time to that firing time. The simulation process continues until the maximum simulation time limit is reached, at which statistics are collected and the simulation is terminated.

4.5. Model validation

Validation is an essential step to ensure that the extracted reliability model accurately reflects the real-world system being modeled. The goal of validation is to ensure that the simulation results are reliable and trustworthy, and that they can be used to make informed decisions.[38] We propose to validate extracted models in two phases: *validation of initial model* and *validation of model at run time* (Figure 12). In the first phase, the validity of a newly extracted model is evaluated to ensure that the model is safe to deploy in a production environment to support decisions. In the second phase, the deployed model is validated to ensure continuous validity. In the following two sections, we elaborate on these two phases in detail.

For both validation phases, input from domain experts, such as production managers or engineers, is essential. This input includes the identification and definition of input and output data streams used for validation, the definition of key performance indicators (KPIs), and the definition of accuracy requirements, frequency of validation, and sufficient data quantity.

Input and output data streams of a manufacturing system typically include a wide range of data types, such as sensor data from machines and other resources, production data, maintenance and repair data, logistics data, and other related data sources. Input data streams include, e.g., orders, production schedules, raw material data, and quality control data, while output data streams include, e.g., material flow, production rates, product quality data, maintenance and repair data, logistics, and supply chain data.

KPIs for a manufacturing system are metrics that are used to measure the performance of the system and can be derived from the production output data and used to assess the validity of the data-driven model. KPIs that can be used for the validation process of data-driven reliability models are, e.g.:

- Production volume: Total amount of production orders completed over a given period of time.
- Throughput: Number of production orders that can be produced or processed by the system over a given period of time.
- Cycle time: Time it takes for a single production order to move through the manufacturing process.
- Work-in-progress (WIP): Amount of inventory in the system at any given time.
- Overall equipment effectiveness (OEE): Measure of how efficiently resources are being used in a production process.
- Resource downtime: Amount of time that resources in the manufacturing system are not operational due to breakdowns, maintenance, or other reasons.

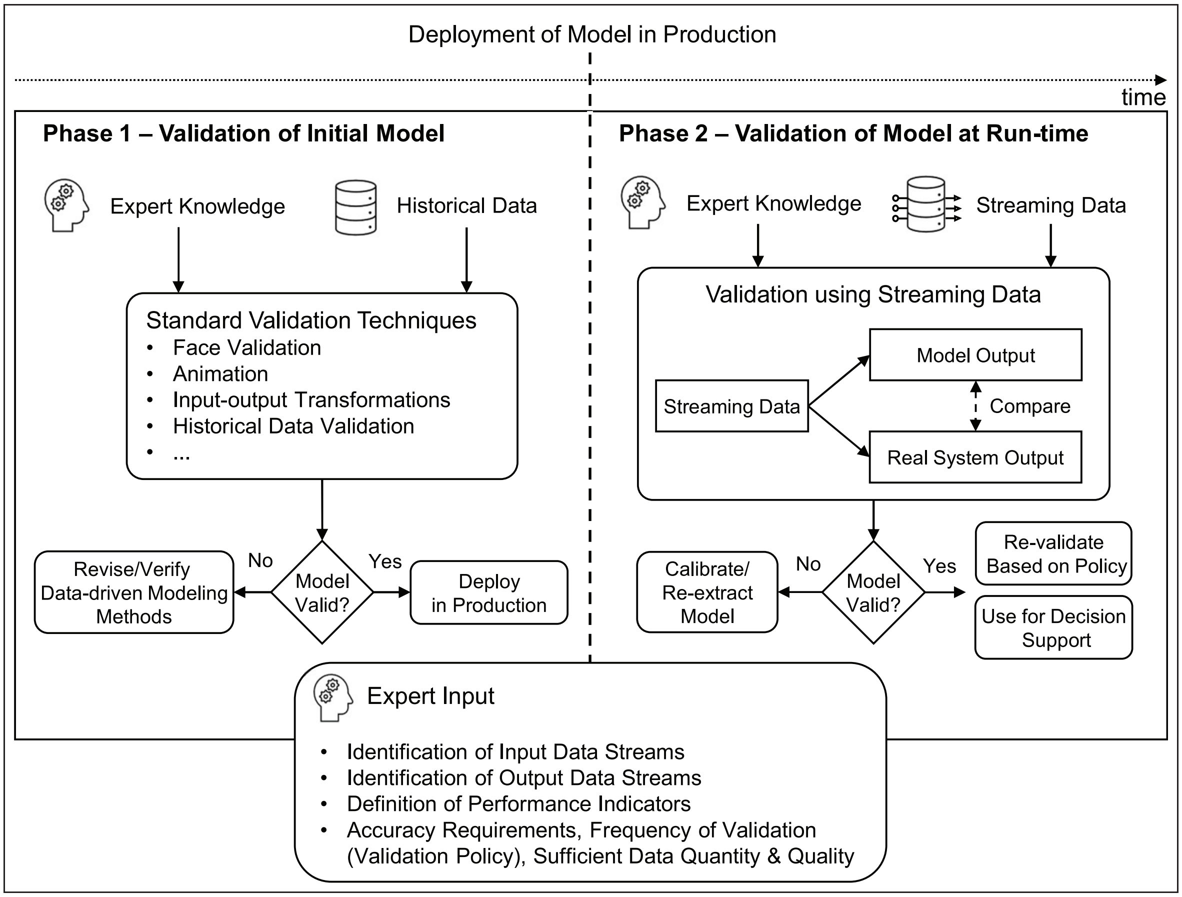

Figure 12. Operational validation of data-driven reliability models in two phases.

Accuracy requirements for data-driven reliability models of manufacturing systems can vary depending on the specific application and purpose of the model. For instance, in some cases, a high degree of accuracy may be necessary to ensure safety or to prevent financial loss, while in others, a lower level of accuracy may be acceptable. Some factors that may affect the required level of accuracy include the complexity of the system, the potential impact of errors, and the criticality of the system. Particular attention must be paid to the environment in which the model is deployed, especially when models are automatically updated in real time, as deviations from the correct model can lead to damaging decisions. In addition, the accuracy requirements may differ between the two validation phases. In the first phase, a higher level of accuracy may be necessary to ensure that the model is safe to deploy, while in the second phase, a lower level of accuracy may be acceptable as long as it does not compromise the safety or efficiency of the system.

The frequency of validation refers to how often a data-driven reliability model which is deployed in production is validated against the real-world system. The choice of validation frequency depends on the criticality of the decision-making that relies on the reliability model, as well as the rate of change in the system being modeled. If the validation frequency is too low, the simulation model may not accurately reflect the current state of the system, potentially leading to incorrect decisions. On the contrary, if the validation frequency is too high, the validation process may become too time-consuming and resource-intensive, resulting in decreased efficiency. In the case of a manufacturing system, where the production process and the machinery involved can change rapidly due to factors such as machine breakdowns or changing production schedules, a higher validation frequency may be necessary. This ensures that the reliability model accurately reflects the current state of the system and that decisions made based on the model are valid. However, a higher validation frequency also requires more data collection and processing, which can be a challenge for large-scale manufacturing systems with high volumes of data. In such cases, it may be necessary to strike a balance between the validation frequency and the available resources for data collection and processing.

Defining sufficient data quantity for validation is an important aspect in ensuring the accuracy of data-driven reliability models of manufacturing systems. If the data quantity is too small, the model may not accurately reflect the system behavior and may lead to incorrect decisions. On the contrary, if the data quantity is too large, it may lead to unnecessary computational costs and delays in decision-making processes. Therefore, it is important to define the appropriate data quantity for validation based on the model's intended purpose, complexity, and the availability of data. In general, the data should be representative of the system behavior and cover a sufficient range of operating conditions. The required data quantity can also vary depending on the frequency of validation. If the model is validated frequently, a smaller data quantity may be sufficient. On the contrary, if the validation frequency is low, a larger data quantity may be necessary to ensure the model's accuracy over a longer period of time.

Finally, the quality requirements of the data used for extracting reliability models need to be determined. Within the PM community, the quality of event and state logs is an ongoing area of research.[39] Data quality encompasses several critical aspects, including accuracy (i.e., the alignment of data with the underlying system), completeness (i.e., the ability to represent all relevant facets of the underlying system), consistency (i.e., adherence to integrity constraints and business rules), and reliability (i.e., the trustworthiness of the data for making informed decisions).[40] Moreover, in the context of PM, specific data quality issues arise, such as distorted attribute value resources (where a resource is referenced in multiple ways in an event log), inadvertent time travel (i.e., incorrect temporal ordering in an event log), and polluted labels (i.e., semantic pollution of activity labels referring to the same logical activity).[39] These considerations underscore the importance of addressing data quality in the context of reliability model extraction.

In general, there are several factors that can make a data-driven reliability model of a manufacturing system invalid. Some examples are as follows:

- Inaccurate or incomplete data: If the input data used to extract the model are inaccurate or incomplete, the resulting model may not accurately represent the actual manufacturing system.
- Insufficient data: If there are not enough data available to extract the model, the resulting model may be incomplete or inaccurate.
- Model overfitting: If the extracted model is overfit to the data it has been extracted from, it may not generalize well to new data and may produce inaccurate results.
- Model complexity: If the model is too complex, it may be difficult to validate and may produce inaccurate results.
- Model assumptions: If the model makes assumptions that do not hold true in the real system, it may produce inaccurate results.
- Changes in the real system: If the real system changes in a way that is not captured by the model, the model may become invalid and produce inaccurate results.

4.4.1. Phase 1—validation of initial model. The first phase of our validation approach for data-driven reliability models of manufacturing systems involves validating the initial model to ensure it is safe for deployment in production to support decisions. This validation process leverages both expert knowledge of the system and historical data to ensure that the model accurately represents the real-world manufacturing system. Standard validation techniques such as face validation, animation, input–output transformation (IOT), and historical data validation (HDV) are utilized during this phase to identify any potential issues or discrepancies with the model. If the model is found to be valid, it can be deployed in production to support decision-making processes. However, if the model is deemed invalid, additional investigations need to be conducted to identify the root cause of the issue. This includes validating and revising the data used to extract the model as well as the generalized conceptual models and design patterns used for implementation of the data-driven modeling methods, and verifying the implementation of these methods.

To conduct an objective comparison between the output of a data-driven reliability model and the actual output of a real system, statistical techniques such as confidence intervals or hypothesis tests can be utilized. These methods enable the comparison of means, variances, or distributions of KPIs or output variables of a reliability model and the real system. In the case of IOT, only the output from the actual system is compared with the output generated by the reliability model, without taking into account real data for the input random variables. On the contrary, when using HDV, both the output and input data from the actual system are incorporated into the evaluation.[41]

To assess the similarity between the outputs of the real system and the reliability model, we calculate confidence intervals using the *t-distribution*. By comparing the KPI of the real system and the confidence interval of the reliability model, we can determine if their outputs are statistically different or not, and thus whether the reliability model accurately captures the behavior of the real system. To do so, let X_{obs} and X_{sim} be the observed and simulated output and let μ_{obs} be the mean of the observed data set and μ_{sim} be the mean of the simulated data set. The confidence intervals for μ_{obs} and μ_{sim} can be calculated as follows:

$$CI_{obs} = [\mu_{obs} - t_{\alpha/2} * SD_{obs}, \mu_{obs} + t_{\alpha/2} * SD_{obs}]$$

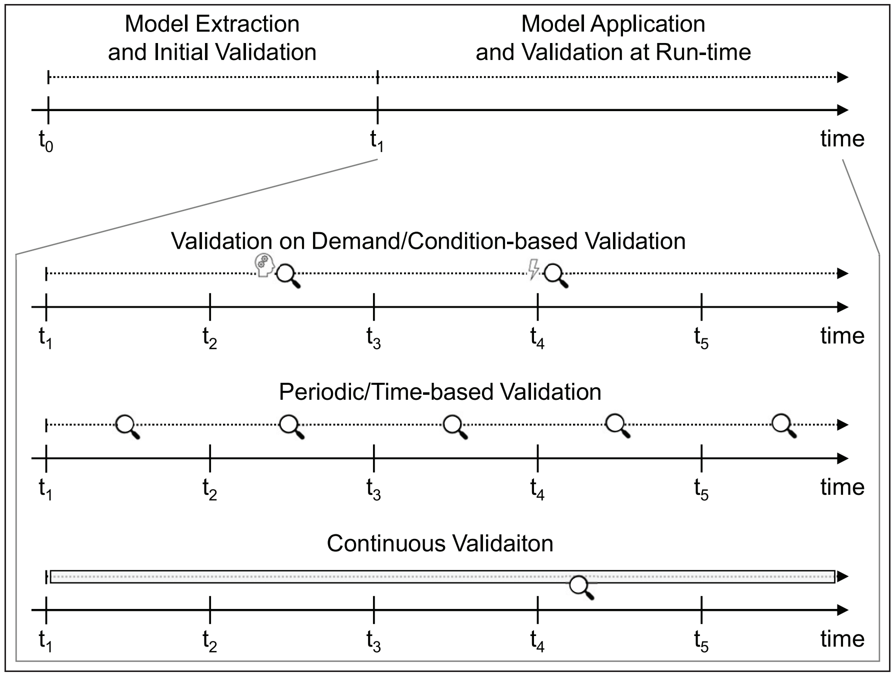

Figure 13. Policies for the validation of a model at run time.

$$CI_{sim} = \left[\mu_{sim} - t_{\frac{\alpha}{2}} * SD_{sim}, \mu_{sim} + t_{\frac{\alpha}{2}} * SD_{sim}\right]$$

where $t_{\alpha/2}$ is the t-value corresponding to the desired level of confidence and the degrees of freedom, and SD_{obs} and SD_{sim} are the standard deviation of the mean of the observed and the simulated data. The degrees of freedom for the *t-distribution* are calculated as $df = n - 1$ where n are the numbers of data points in the observed, respectively, simulated data.

There are various ways to assess whether the reliability model accurately represents the real system. If sufficient data are available for both the observed (X_{obs}) and simulated (X_{sim}) values, confidence intervals can be constructed and compared. If the confidence intervals overlap, this suggests that the means of the observed and simulated data sets are not significantly different. However, there may be cases where only one value, such as x_{obs} or μ_{obs}, is known. In such cases, if this value falls within the confidence interval of the simulated data (CI_{sim}), it can be assumed that the reliability model is not significantly different from the real system. However, such a model may not be very robust.

4.4.2. Phase 2—validation of model at run time. The second phase of our validation approach for data-driven reliability models of manufacturing systems is the validation of the model at run time. This phase ensures the ongoing validation of the deployed model. During this phase, the output of the model is compared with the actual output of the real system based on data streams collected during the production process. Any discrepancies found are rectified by adjusting the model parameters, inputs, or logic. These adjustments may be automatic or require intervention from domain experts. In addition, the validation of the model at run time may involve updating the model to account for changes in the production process, such as changes in the environment or equipment. This can be done by either calibrating the model parameters or by re-extracting the model.

There are three policies according to which the validation of a data-driven simulation model at run time is ensured (Figure 13):

- Validation on demand/condition-based validation: An expert determines the best time to validate the model based on domain knowledge or when the validation is triggered based on the condition of the system or its components.
- Periodic/time-based validation: A domain expert determines a specific schedule for model validation.
- Continuous validation: The model is continuously validated.

These policies have different computational complexities, so it is crucial to determine the appropriate policy for the system or component being validated.

Similar to the first validation phase, an objective comparison between the output of a data-driven reliability model and the actual output of a real system can be

Table 2. Decisions that can be supported using the extracted reliability model.

Decision type	Decision	Description	Necessary adjustments in the reliability model
System configuration/ purchase decisions	Resource efficiency	Increasing/decreasing resource efficiency	Adjust distribution/sampled values of resource activity transition
	Resource reliability	Increasing/decreasing resource reliability	Adjust distribution/sampled values of resource failure transition
	Resource redundancy	Adding/removing a resource	Add/remove the places and transition referring to a resource
	Routing	Increasing/decreasing the utilization of a resource	Adjust firing probability of immediate transitions representing routing logic
Maintenance	Preventive maintenance	Introducing preventive maintenance	Reset firing time of resource failure transition after specified time
	Maintenance staffing	Better/more maintenance crews	Adjust distribution/sampled values of resource repair transition
	Maintenance prioritization	Prioritization of resource repairs	Add model components that reflect the prioritization
Scheduling	Order scheduling	Improving order schedules	Replace distribution of arrival transition with pre-defined schedule

conducted using statistical techniques such as confidence intervals or hypothesis tests. Furthermore, in highly dynamic systems like manufacturing systems, the observation window plays a critical role in determining the accuracy of the validation results. For instance, in a flexible production line where machines switch between producing different products, the activity times of machines may vary, resulting in different cycle times for different products. Thus, to ensure the accuracy of a data-driven reliability model, the validation run should focus on data collected during the production of a specific product, depending on the required level of detail for the model.

4.6. Model application for decision support

After validating an extracted reliability model, it can be used to inform various kinds of decisions. Table 2 provides examples of the types of decisions that can be supported, which include system configuration, maintenance planning, and scheduling. To make use of the model, adjustments may be needed, such as modifying distribution functions for timed transitions, updating firing probabilities for immediate transitions, or introducing new model components.

After modifications have been applied to a model, it is simulated to assess the impact on the system. Similar to the validation process, this involves selecting an appropriate KPI that can be used to compare the model before and after modifications have been applied. For instance, the KPI *production volume* would be suitable for evaluating the impact of changes in equipment cycle time on the system. To obtain accurate results, multiple replications of the simulation experiment should be run. This allows to account for random variation in the system's behavior and obtain a more robust estimate of the impact of the modifications.

5. Case study

In this section, we present two case studies demonstrating the application of our proposed framework for data-driven reliability assessment of manufacturing systems. The first case study is based on the *Industry 4.0 Lab* (I4.0 Lab) at the University of Southern Denmark and demonstrates the model extraction, initial model validation as well as model application aspects of our framework. In the second case study, we illustratively demonstrate the validation of extracted reliability models at run time.

5.1. Case Study 1—model extraction, initial validation, and application

The *I4.0 lab* is a strategic initiative that targets the support of research, industry collaboration, innovation, and education in the latest I4.0 technologies.[42,43] The *I4.0 lab* integrates knowledge from different research groups of the university and production resources from different technology providers. An ongoing iterative project was started to gain knowledge and experience with these resources. The *I4.0 lab* included a production system that produced a

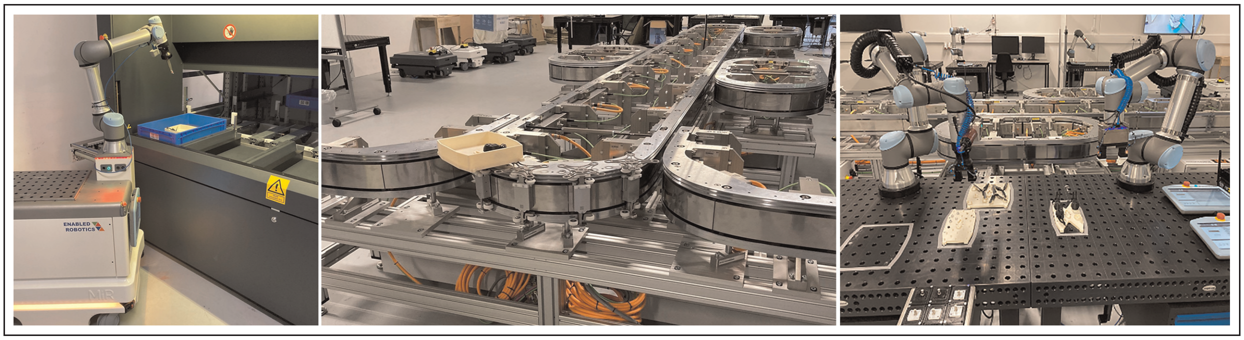

Figure 14. Production resources in the *I4.0 lab*. (Left). An Automated Guided Vehicle picks up drone parts from a warehouse, (middle) a magnetic transport track transports drone parts using shuttles, and (right) a production cell assembles landing gear on a drone motor.

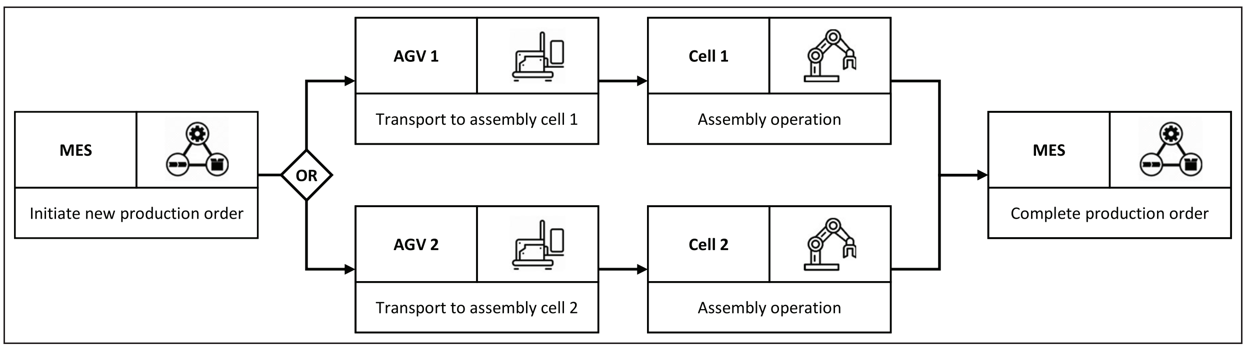

Figure 15. Overview of the case study system.

simplified drone. Figure 14 illustrates an excerpt of the resources involved in the production sequence, which we describe in detail in the following.

The case study in this section is inspired by the resources and production process in the *I4.0 lab*. In the following subsections, we first describe the case study system and the data we extracted from it, followed by the data-driven model extraction and validation process. Finally, we demonstrate the utilization of the extracted model to aid decision-making in maintenance staffing.

5.1.1. Case study system and extracted data. Figure 15 provides an overview of the case study system, which is a flow production line commonly found in manufacturing systems. The production line is fully automated and consists of five resource components: an MES, two AGVs, and two assembly cells. Both assembly cells work concurrently, performing the same assembly operation. The MES controls the production process by initiating new production orders, directing them to either assembly cell 1 or 2, and marking orders as completed. When a new production order is initiated and assigned to one of the assembly cells, the AGVs transport the raw material to the designated cell.

The AGVs and the assembly cells are susceptible to failures, while the MES is always fully operational. In the event of a production resource failure, the resource stops operating and a repair crew is dispatched to repair the malfunctioning resource. The maintenance policy in place is purely reactive. The AGV has an unlimited buffer and a capacity of one, while both assembly cells 1 and 2 have a finite buffer and a capacity of one.

To conduct experiments, we developed a simulation model (Figure 20 in Appendix) using *AnyLogic*[44] based on the described case study system. We used this simulation model to generate synthetic data consisting of an event log and a state log. These synthetic data reflect data that can be collected in real manufacturing systems. The generated event and state logs capture the production of 4.431 orders over a 1-month time span. An excerpt of the generated event and state logs is displayed in Tables 3 and 4, respectively.

5.2. Reliability model extraction

Based on the event and state log, we extracted a reliability model employing our proposed approach for data-driven reliability modeling. Figure 16 depicts the extracted reliability model. We successfully extracted and parameterized the manufacturing process model and the resource fault models. Each timed transition is associated with its corresponding distribution function, while immediate transitions display their respective weights. In addition, we have

Table 3. Excerpt of the generated event log.

Timestamp	Order ID	Resource	Event	Event type
01-06-2023 00:00:05	442	mes	new_order	NA
01-06-2023 00:00:05	442	mes	direct_to_line1	NA
01-06-2023 00:00:05	442	agv1	agv1_transport_to_cell1_buffer	start
01-06-2023 00:04:08	442	agv1	agv1_transport_to_cell1_buffer	end
01-06-2023 00:04:08	442	mes	enter_cell1	NA
01-06-2023 00:04:08	442	cell1	cell1_operation	start
01-06-2023 00:07:39	443	mes	new_order	NA
01-06-2023 00:07:39	443	mes	direct_to_line1	NA
01-06-2023 00:07:39	443	agv1	agv1_transport_to_cell1_buffer	start
01-06-2023 00:10:29	443	agv1	agv1_transport_to_cell1_buffer	end
01-06-2023 00:11:53	442	cell1	cell1_operation	end
01-06-2023 00:11:53	442	mes	order_completed	NA
01-06-2023 00:11:53	443	mes	enter_cell1	NA
...	...	...	...	...

Table 4. Excerpt of the generated state log.

Timestamp	Resource	State
01-06-2023 00:00:05	agv1	busy
01-06-2023 00:04:08	agv1	idle
01-06-2023 00:04:08	cell1	busy
01-06-2023 00:07:39	agv1	busy
01-06-2023 00:10:29	agv1	idle
01-06-2023 00:11:53	cell1	idle
01-06-2023 00:11:53	cell1	busy
01-06-2023 00:18:05	agv1	busy
01-06-2023 00:20:15	cell1	idle
01-06-2023 00:21:09	agv1	idle
01-06-2023 00:21:09	cell1	busy
01-06-2023 00:21:31	agv1	busy
01-06-2023 00:25:07	agv2	busy
...	...	...

extracted the capacity of both assembly cells, along with their finite buffer sizes.

In the Appendix, we show the histograms and fitted distribution functions (Figures 21–24 in Appendix) as well as the graphical representation of the extracted model using the visualization functionality of our data-driven reliability assessment tool (Figure 25 in Appendix).

5.3. Simulation and validation of the extracted model

We simulate the extracted reliability model using DES and validate the model using our approach described previously in this article. We use *production volume* and *resource downtime* as the KPIs to compare the outputs of the real system with the outputs of the reliability model.

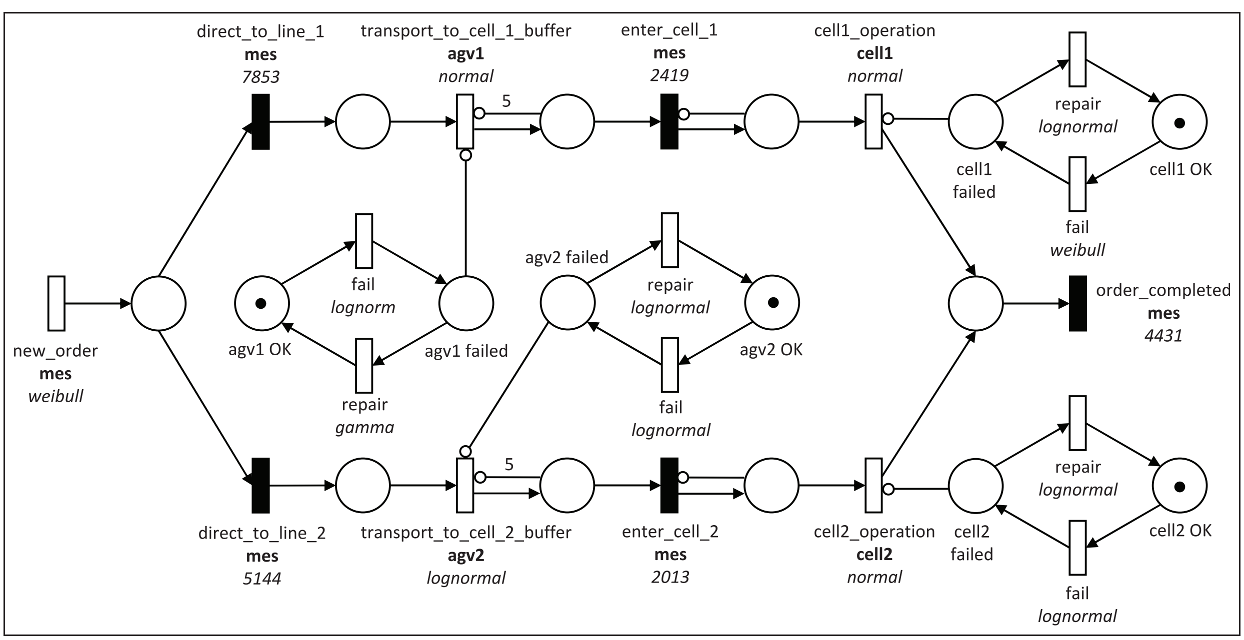

Figure 16. Extracted reliability model of the case study system.

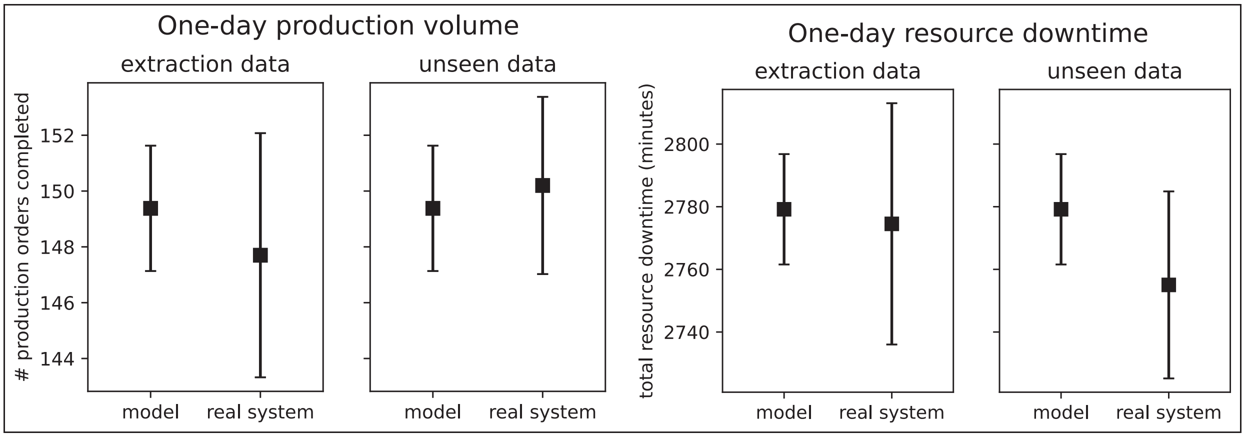

Figure 17. Validation results.

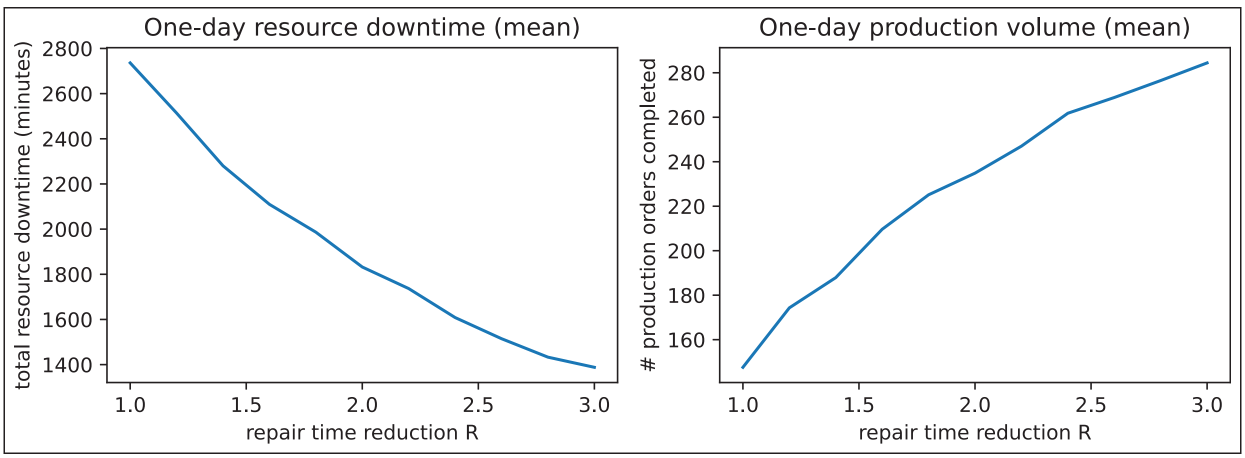

Figure 18. Effect of reducing the repair time duration of production resources.

Figure 17 illustrates the validation results for our extracted reliability model using IOT after 100 simulation replications at a confidence level of 95%. We validate the model using the same data the model was extracted from, as well as an unseen data set capturing the production system during another month. As depicted, the confidence intervals of the model and the real system overlap for both KPIs, representing the production volume over 1 day and the total downtime of all production resources over 1 day. Consequently, we can assume the extracted model to be valid for the system under study.

5.4. Model application and decision support

To demonstrate the application of the previously extracted and validated reliability model, we consider the following scenario. The production manager wants to analyze how reducing the repair time for production resources will affect both the total resource downtime and the production volume. Based on this analysis, the production manager can make informed decisions, such as investing in better training for existing repair crews or determining the need for additional repair crews.

We test this scenario by adjusting the distribution functions of the repair transitions of the production resources in the SPN. Specifically, we decrease the duration of the repair activities by a factor ranging from 1 to 3 with a step of 0.1. A reduction factor of 3 implies that the repair activities are performed three times faster compared with the original configuration. Mathematically, the reduction factor R can be described as $R = \frac{T_o}{T_r}$ where T_o is the original repair duration and T_r is the reduced repair duration. Figure 18 demonstrates the significant impact of reducing the repair duration on both resource downtime and production volume. The figure presents the mean values of both KPIs after conducting 100 simulation replications for each reduction factor.

5.5. Case Study 2—validation at run time

In this case study, we demonstrate how to validate an extracted reliability model after it has been deployed in

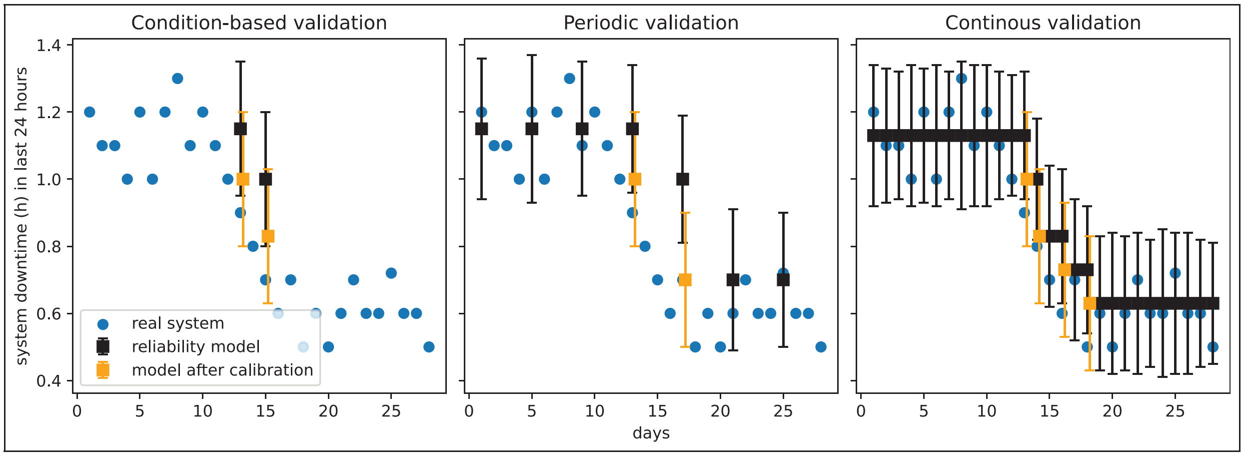

Figure 19. Validation and calibration of the reliability model following the three validation policies.

production. Note that this case study is for illustrative purposes only and the functionality has not yet been fully implemented in our reliability assessment tool.

Based on an extracted reliability model, we applied the proposed validation policies at model run time. We streamed 30 days of production data recorded in the event and state logs to emulate a live production scenario. Our validation KPI was the *total system downtime (in hours)* in the last 24 h, which we calculated at the end of each day. Figure 19 displays the results obtained from applying the different validation policies. For condition-based validation, we set a trigger to validate the model if a decline in the KPI was observed over the past 3 days. For periodic validation, we scheduled the validation to occur every fourth day. For continuous validation, we validated the model each time a new KPI measurement became available. Note that there was a significant decrease in the KPI between days 10 and 15, as depicted in the Figure.

We conducted 50 simulation replications for each model validation iteration to calculate the confidence intervals depicted. When the recorded KPI was outside the confidence interval bounds, we re-calibrated the reliability model using data from the past 24 h. As expected, the continuous validation policy provided the best results and kept the reliability model consistently accurate. We also found that the periodic validation policy kept the model reasonably accurate. In contrast, the condition-based validation policy struggled to capture the latest changes in the real system accurately.

6. Summary and discussion

The reliable operation of complex manufacturing systems is crucial to avoid production disruptions, expensive downtime, and potential safety issues. However, the increasing complexity and dynamic nature of manufacturing systems requires a more data-driven and less labor-intensive approach to reliability assessment to accurately represent and analyze the behavior of these systems. To address these challenges, we proposed a novel framework for data-driven hardware reliability assessment of manufacturing systems using PM in this article.

Our framework consists of three main components: (1) reliability model extraction, (2) simulation and validation, and (3) application. For the first component, our approach takes an event log and a state log as input, and uses PM and statistical analysis techniques such as MLE to extract a reliability model of a manufacturing system using SPNs as the modeling formalism. By extracting and integrating material flow, timing information, resource capacities, and resource fault models, our approach extracts comprehensive simulation models of reliability-focused manufacturing system behavior.

For the model simulation and validation component, we proposed a two-phase approach. In the first phase, *validation of initial model*, the validity of a newly extracted reliability model is evaluated to ensure that the model is safe to deploy in a production environment to support decisions. We described several KPIs, such as production volume and resource downtime, which can be used to compare the output of the real system with the output of the reliability model simulation. In the second phase, *validation of model at run time*, the deployed model is validated to ensure ongoing validity in case of changes in the physical system. For the validation of a model at runtime, we suggest three validation policies: condition-based validation, periodic validation, and continuous validation.

For the model application component, we addressed how an extracted and validated model can be applied to support decision-making. Models can be modified by adjusting model parameters or introducing/removing components and then simulated using DES to assess the impact of these modifications on the system. The results of these simulations can then be used to support decisions regarding system configuration, maintenance, and scheduling.

To demonstrate the application of our developed framework, we conducted two case studies. For the first case study, we applied our framework to a flow line. For the second case study, we demonstrated the validation of a model at run time, following our proposed model validation approach.

In summary, our developed framework for data-driven reliability assessment of manufacturing systems offers several advantages over traditional methods. First, extracted reliability models can be automatically kept up-to-date in real time. When the configuration or topology of the actual system changes, the extracted reliability model can either be re-extracted or calibrated using the latest data from the system. In contrast, traditional expert-based reliability models require manual updates, which can become a significant bottleneck, especially for complex and highly dynamic manufacturing systems. Second, our framework is a generic approach and can be used to extract reliability models for a wide array of manufacturing systems, as long as the required data for model extraction and validation (i.e., event and state logs) are available. Traditional methods, on the contrary, often demand custom development of reliability models for each manufacturing system under study. Finally, by using a data-driven approach, we can reveal differences between the initial system design or a manually developed model, and the actual process that takes place in a manufacturing system. Data can support active monitoring of conformance between the modeled behavior and actual behavior of a system (i.e., conformance checking) and update the model if necessary.

7. Limitations and future work

Our framework for data-driven reliability assessment of manufacturing systems has a number of limitations that call for future research.

First, we focus mainly on hardware production resources in our framework. However, besides hardware, a manufacturing system consists of software as well has humans. In future, we plan to incorporate software and human reliability information into our framework as well.

Second, we employed inhibitor arcs to model resource capacities in SPNs. While they are commonly used for this purpose and offer a straightforward approach to modeling capacities, they can render formal network analysis quite challenging. In future research, we intend to explore the use of additional places to represent resource capacities.

Third, we assume that process discovery algorithms such as the *α -miner* algorithm,[33] *HeuristicsMiner*[34] or the *Inductive Miner*[35] yield an accurate model of the underlying system, which can then be used to extract the manufacturing process model. However, it is important to note that there is no guarantee of extracting an accurate model with these algorithms as they all have individual shortcomings. Moreover, future work is needed to evaluate the most suitable modeling formalisms (i.e., SPNs and C-nets) to represent the behavior of real manufacturing systems.

Fourth, we do not specify methods for integrating information from condition monitoring data into a data-driven reliability model. Event detection in condition monitoring data, e.g., using machine learning techniques, could improve the level of detail and accuracy of extracted fault models of production resources.

Fifth, we assume that we have sufficient data on faults and repairs for the extraction of reliability models available, which are inherently uncommon events when compared to more frequent events such as resource activities (e.g., assembly operations and transport of goods). The integration of IoT sensors in modern manufacturing systems, however, is enabling systematic and large-scale collection of fault and repair data. Moreover, in manufacturing systems, instances of failure are generally more frequent compared with safety-critical systems such as aircraft or nuclear power plants. In future research, we also aim to investigate the integration of expert knowledge in the model extraction procedure. For example, when extracting a model for a new production line or process, the extracted model can be augmented with expert knowledge for rare events. Once streaming data of the system capturing rare events are available, the model can be further calibrated.

Sixth, in our framework, we do not distinguish between different types of products being processed in a manufacturing system. However, in many manufacturing systems, multiple types of products are processed in the same system. To address this limitation, we plan to introduce colored tokens similar to those used in colored PNs. By using colored tokens, we can extract different possible paths of production orders through the system and then model them with a (colored) SPN. We expect that this will increase the accuracy of the reliability model.

Seventh, we assume that the data used to extract reliability models are valid and of good quality, as evaluating these aspects was beyond the scope of the article. This reliance on data validity and quality in turn may make our framework vulnerable to dealing with, e.g., missing data or extreme values.

Finally, we plan to extend the model application component for decision support in future research. Currently, decisions regarding system configuration, maintenance, and scheduling can be supported using extracted reliability models. We intend to extend the extracted models to support additional decisions. We, furthermore, plan to develop a framework for component-based data-driven Digital Twins, where the type of decision at hand defines the model components to be extracted. Notably, we have already initiated research in this direction by Friederich et al.[27]

The case studies we conducted also have a number of limitations. In the first case study, we only use synthetic data generated by a simulation model to extract and validate a reliability model. The synthetic data are an idealized representation of the data that can be collected in real manufacturing systems. In the second case study, we demonstrated our validation approach for models during run time in an illustrative case study where we also used synthetic data. Through our research, we discovered a lack of reference models and benchmarks for reliability model extraction and validation. Thus, we see this important topic as part of our future work. In addition, the manufacturing processes considered in our case studies are not very complex. Therefore, the applicability of our framework to larger and more complex systems or processes has yet to be proven. As a result, we plan to further develop our approach in the future and test it on more complex systems.

In summary, in future work, we plan to further develop our framework to include all three resource types (i.e., hardware, software, and humans) of manufacturing systems and to integrate both data and expert knowledge into the model extraction process. In addition, we plan to integrate more data on faults of production resources and increase the level of detail of the extracted reliability models—especially the fault models. We also plan to test our framework on more case studies, especially using real data from production systems, and to evaluate the applicability of our framework to more complex systems than those covered in this article.

Appendix

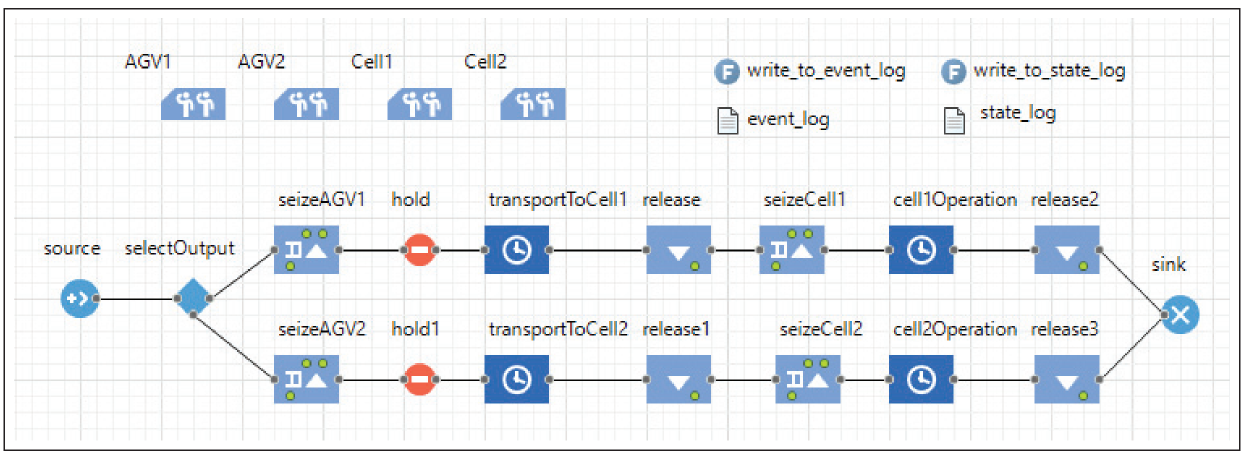

Figure 20. *AnyLogic*[44] model used to generate synthetic data of the case study system.

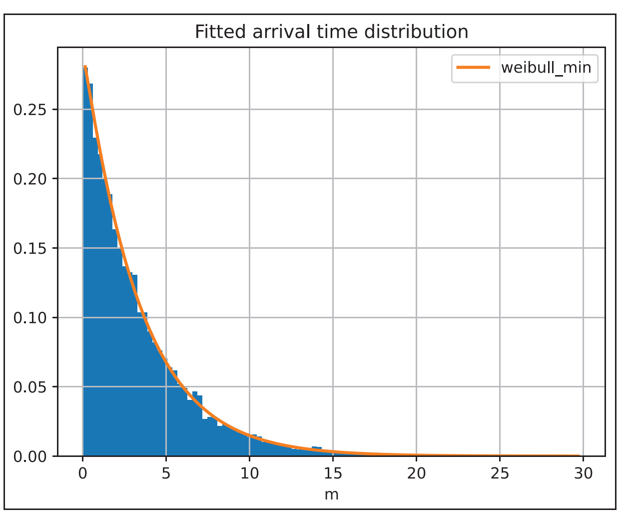

Figure 21. Fitted arrival time distribution.

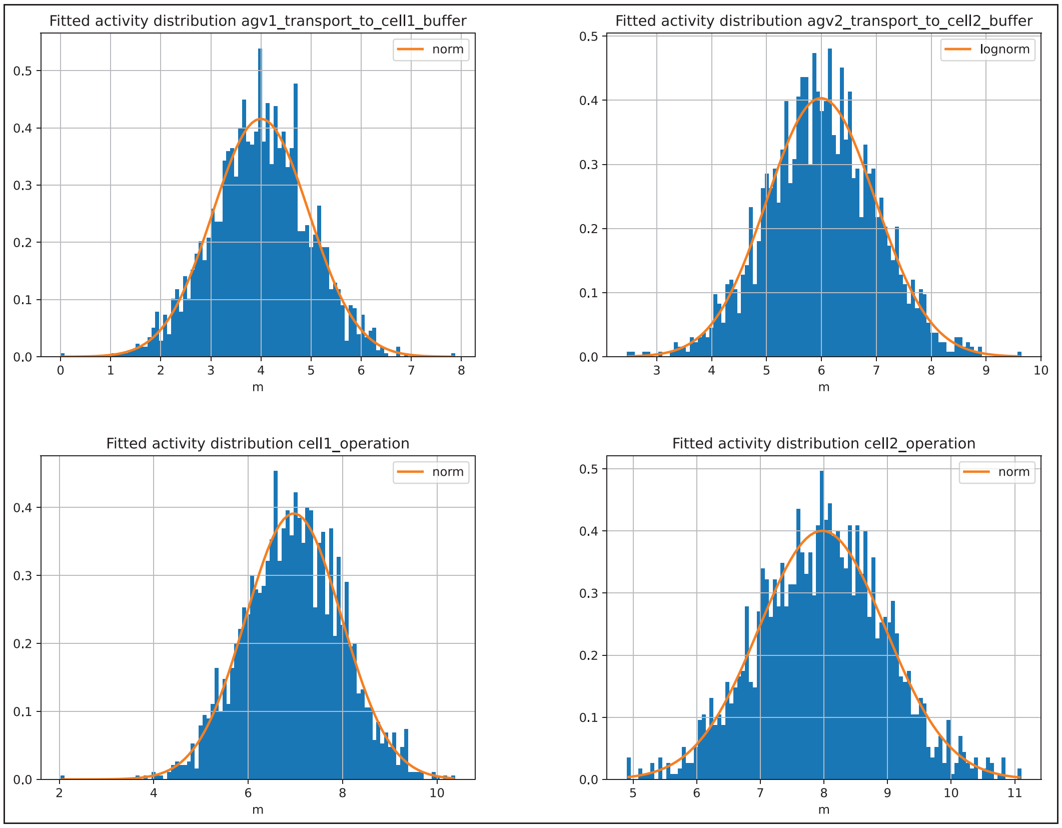

Figure 22. Fitted resource activity distributions.

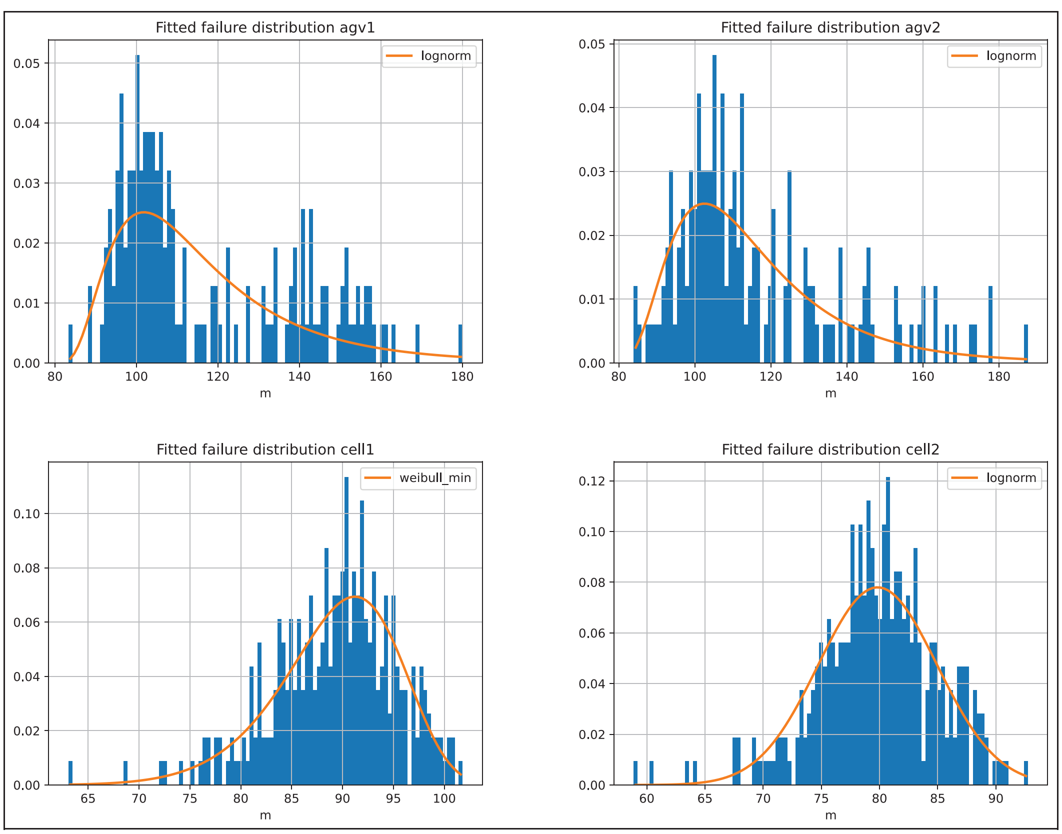

Figure 23. Fitted resource failure distributions.

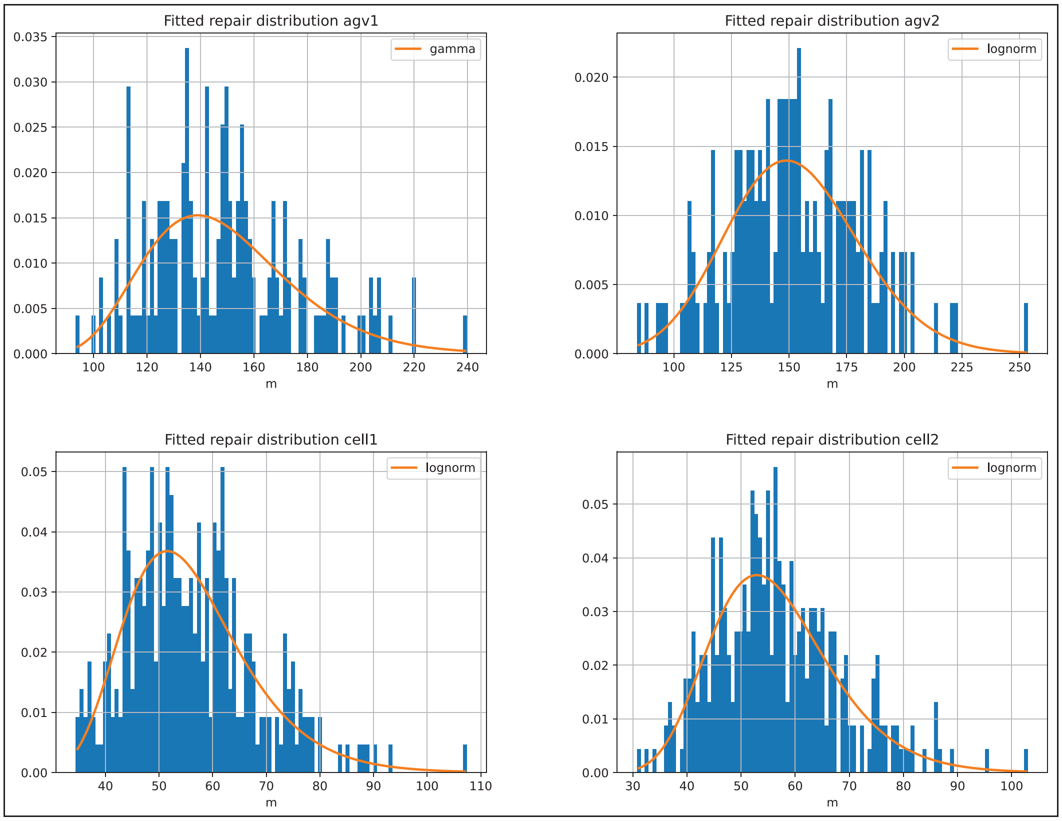

Figure 24. Fitted resource repair distributions.

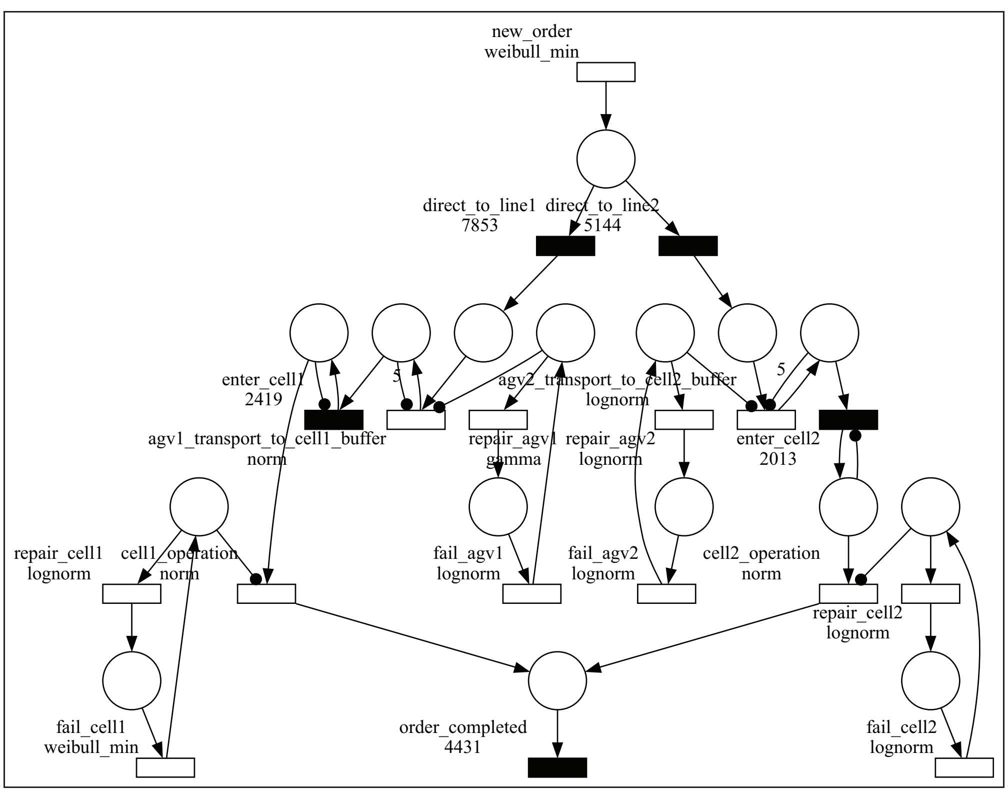

Figure 25. Graphical representation of the extracted model.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

ORCID iD

Jonas Friederich https://orcid.org/0000-0001-9034-5907

References

1. Lasi H, Fettke P, Kemper HG, et al. Industry 4.0. *Bus Inf Syst Eng* 2014; 6: 239–242.
2. Monostori L. Cyber-physical production systems: roots, expectations and R&D challenges. *Procedia CIRP* 2014; 17: 9–13.
3. Lazarova-Molnar S, Mohamed N and Shaker HR. Reliability modeling of cyber-physical systems: a holistic overview and challenges. In: *2017 workshop on modeling and simulation of cyber-physical energy systems (MSCPES)*, Pittsburgh, PA, 21 April 2017. New York: IEEE, pp. 1–6.
4. Blischke WR and Murthy DNP. *Reliability: modeling, prediction, and optimization*. Cham: John Wiley, 2011.
5. Bazovsky I. *Reliability theory and practice*. Chelmsford, MA: Courier Corporation, 2004.
6. Dai P, Yang L, Zeng Y, et al. Data-driven reliability evaluation of the integrated energy system considering optimal service restoration. *Front Energy Res* 2022; 10: 934774.
7. Bourouni K. Availability assessment of a reverse osmosis plant: comparison between reliability block diagram and fault tree analysis methods. *Desalination* 2013; 313: 66–76.
8. Kim H and Singh C. Reliability modeling and simulation in power systems with aging characteristics. *IEEE Trans Power Syst* 2009; 25: 21–28.
9. Wang W, Reddy V, Krishnan AT, et al. Compact modeling and simulation of circuit reliability for 65-nm CMOS technology. *IEEE Trans Device Mater Rel* 2007; 7: 509–517.
10. Lazarova-Molnar S and Mohamed N. Reliability assessment in the context of industry 4.0: data as a game changer. *Procedia Comput Sci* 2019; 151: 691–698.
11. Friederich J and Lazarova-Molnar S. Process mining for reliability modeling of manufacturing systems with limited data availability. In: *2021 8th international conference on internet of things: systems, management and security (IOTSMS)*, Gandia, 6–9 December 2021, pp. 1–7. New York: IEEE.
12. Friederich J and Lazarova-Molnar S. Towards data-driven reliability modeling for cyber-physical production systems. *Procedia Comput Sci* 2021; 184: 589–596.
13. van der Aalst W. Data science in action. In: van der Aalst WMP (ed.) *Process mining: data science in action*. Berlin: Springer, 2016, pp. 3–23.
14. Ferronato JJ and Scalabrin EE. PM2Sim: the automated creation of a simulation model from process mining. In: *2021 IEEE international conference on systems, man, and cybernetics (SMC)*, Melbourne, VIC, Australia, 17–20 October 2021, pp. 2232–2237. New York: IEEE.
15. Lugaresi G and Matta A. Automated manufacturing system discovery and digital twin generation. *J Manuf Syst* 2021; 59: 51–66.
16. Kumar A and Lata S. Reliability analysis of piston manufacturing system. *J Reliab Statist Stud* 2011; 4: 43–55.
17. Myszewski JM. Mathematical model of the occurrence of human error in manufacturing processes. *Qual Reliab Eng Int* 2010; 26: 845–851.
18. Lazarova-Molnar S, Niloofar P and Barta GK. Data-driven fault tree modeling for reliability assessment of cyber-physical systems. In: *2020 winter simulation conference (WSC)*, Orlando, FL, 14–18 December 2020, pp. 2719–2730. New York: IEEE.
19. Volovoi V. Modeling of system reliability Petri nets with aging tokens. *Reliab Eng Syst Saf* 2004; 84: 149–161.
20. Zschech P. A taxonomy of recurring data analysis problems in maintenance analytics. In: *ECIS 2018 proceedings*, Portsmouth, 23–28 June 2018.
21. Jardine AKS, Lin D and Banjevic D. A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mech Syst Signal Process* 2006; 20: 1483–1510.
22. Wang J, Chang Q, Xiao G, et al. Data driven production modeling and simulation of complex automobile general assembly plant. *Comput Ind* 2011; 62: 765–775.
23. Zou J, Chang Q, Arinez J, et al. Data-driven modeling and real-time distributed control for energy efficient manufacturing systems. *Energy* 2017; 127: 247–257.
24. Resman M, Protner J, Simic M, et al. A five-step approach to planning data-driven digital twins for discrete manufacturing systems. *Appl Sci* 2021; 11: 3639.
25. Milde M and Reinhart G. Automated model development and parametrization of material flow simulations. In: *2019 Winter simulation conference (WSC)*, National Harbor, MD, 8–11 December 2019. New York: IEEE.
26. Friederich J, Cai W, Gan BP, et al. Equipment-centric data-driven reliability assessment of complex manufacturing systems. In: *Proceedings of the 2023 ACM SIGSIM conference on principles of advanced discrete simulation. SIGSIM-PADS '23*, Orlando, FL, 21–23 June 2023, pp. 62–72. New York: Association for Computing Machinery.
27. Friederich J, Lugaresi G, Lazarova-Molnar S, et al. Process mining for dynamic modeling of smart manufacturing systems: data requirements. *Procedia CIRP* 2022; 107: 546–551.
28. Friederich J, Francis DP, Lazarova-Molnar S, et al. A framework for data-driven digital twins of smart manufacturing systems. *Comput Ind* 2022; 136: 103586.
29. Niloofar P and Lazarova-Molnar S. Fusion of data and expert knowledge for fault tree reliability analysis of cyber-physical systems. In: *2021 5th international conference on system reliability and safety (ICSRS)*, Palermo, 24–26 November 2021, pp. 92–97. New York: IEEE.
30. Niloofar P and Lazarova-Molnar S. Collaborative data-driven reliability analysis of multi-state fault trees. *Proc Inst Mech Eng, Part O: J Risk Reliab* 2023; 237: 886–896.
31. Friederich J and Lazarova-Molnar S. Data-driven reliability modeling of smart manufacturing systems using process mining. In: *2022 winter simulation conference (WSC)*, Singapore, 11–14 December 2022, pp. 2534–2545. New York: IEEE.
32. Lazarova-Molnar S. *The proxel-based method: formalisation, analysis and applications*. PhD Thesis, Otto-von-

Guericke-Universität Magdeburg, Universitätsbibliothek, 2005.
33. van der Aalst W, Weijters T and Maruster L. Workflow mining: discovering process models from event logs. *IEEE Trans Knowl Data Eng* 2004; 16: 1128–1142.
34. Weijters A, van der Aalst WMP and Alves De Medeiros A. *Process mining with the HeuristicsMiner algorithm*. BETA publicatie working papers, Eindhoven: Technische Universiteit Eindhoven, 2006.
35. Leemans S, Fahland D and van der Aalst WMP. Process and deviation exploration with Inductive visual Miner. In: Limonad L and Weber B (eds) *BPM Demo Sessions 2014 (co-located with BPM 2014), CEUR Workshop Proceedings*, Eindhoven, The Netherlands, 20 September 2014, pp. 46–50. Available at: https://ceur-ws.org/
36. Myung IJ. Tutorial on maximum likelihood estimation. *J Math Psych* 2003; 47: 90–100.
37. Haas P. Stochastic Petri nets for modelling and simulation. In: *Proceedings of the 2004 winter simulation conference*, Washington, DC, 5–8 December 2004, vol. 1. p. 112. New York: IEEE.
38. Sargent RG. Verification and validation of simulation models. In: *Proceedings of the 2010 winter simulation conference*, Baltimore, MD, 5–8 December 2010, pp. 166–183. New York: IEEE.
39. Ter Hofstede AHM, Koschmider A, Marrella A, et al. Process-data quality: the true frontier of process mining. *J Data Inf Qual* 2023; 15: 1–21.
40. Batini C and Scannapieco M. *Data and information quality: dimensions, principles and techniques*. Cham: Springer, 2018.
41. Banks J, Carson J, Nelson B, et al. *Discrete-event system simulation*. Hoboken, NJ: Prentice Hall, 2010.
42. SDU. SDU Industry 4.0 Lab, 2023, https://www.sdu.dk/en/forskning/i40lab (accessed Jul 2023).
43. Jepsen SC, Worm T, Johansen A, et al. A research setup demonstrating flexible industry 4.0 production. In: *2021 international symposium ELMAR*, Zadar, Croatia, 13–15 September 2021, pp. 143–150. New York: IEEE.
44. AnyLogic. AnyLogic: simulation modeling software tools & solutions for business, 2023, https://www.anylogic.com/ (accessed July 2023).

Author biographies

Jonas Friederich holds a PhD in Computer Science from the University of Southern Denmark. His research interests cover Modeling and Simulation, Process Mining, Machine Learning, and Computer Vision.

Sanja Lazarova-Molnar holds two full professorships, at the Karlsruhe Institute of Technology, and the University of Southern Denmark. Her research focuses on data-driven simulation, digital twins, and cyber-physical systems modeling for reliability and energy efficiency enhancement. She leads activities focused on digital twins and data-driven simulation in several European and national projects. Furthermore, she assumes leadership roles in IEEE and represents The Society for Modeling & Simulation International (SCS) to the Winter Simulation Conference (WSC) Board of Directors, contributing significantly to these professional organizations. She was also one of the Proceedings Editors for the WSC in 2019 and 2020 and an associate editor of *SIMULATION: Transactions of The Society for Modeling and Simulation International*.