Journal of Intelligent Manufacturing
https://doi.org/10.1007/s10845-025-02572-x

Real-to-sim: automatic simulation model generation for a digital twin in semiconductor manufacturing

Sebastian Behrendt[1] · **Thomas Altenmüller**[2] · **Marvin Carl May**[1] · **Andreas Kuhnle**[1] · **Gisela Lanza**[1]

Received: 23 February 2023 / Accepted: 17 January 2025

Abstract
Semiconductor manufacturing systems are highly complex due to intricate processes and material flows. Operating these systems efficiently remains a significant challenge, particularly under the growing demands for operational excellence and cost reduction. Current approaches often rely on extensive manual modeling, which slows down production planning and adaptation. To address these challenges, we propose a data-driven methodology for Automatic Simulation Model Generation (ASMG), enhanced by machine learning techniques. This fully automated pipeline extracts and processes production data (lot tracking information and resource states) to generate simulation models without manual intervention. A machine learning technique called equipment emulation captures complex tool behaviors and mitigates issues with noisy or incomplete data. Validation in two real-world semiconductor production environments, covering over 300 days and showing an accuracy within 5–7% for throughput and uptime, demonstrates the method's ability to produce precise models. By reducing the time and expertise required for model creation, this ASMG method facilitates agile digital twin implementations and enables faster, more responsive production planning.

Keywords Digital twin · Machine learning · Automated modeling · Semiconductor manufacturing · Data mining · Simulation

Introduction

Modern production systems are characterized by increasing complexity as a result of shorter product life cycles, rising customer demands and higher product diversity (Wuest et al., 2016). A prominent example is the semiconductor manufacturing industry, which features not only high quality requirements on processes, but also unique complexities in process and material flows (Lingitz et al., 2018). Former factors for cost reduction in semiconductor manufacturing are plentiful, i.e. miniaturization, increased wafer size and improved yield, but decreasingly effective (Mönch et al., 2012). Therefore, operational excellence is a key criterion in the semiconductor industry to maintain an economically efficient production and gain a competitive edge (Mönch et al., 2011). However, achieving operational excellence in semiconductor manufacturing faces significant challenges caused by the intrinsic complexity of associated manufacturing systems (Kuhnle, 2020).

Semiconductor manufacturing systems can be described as complex job shops (Waschneck et al., 2016). Complex job shops pose additional challenges to the typical challenges of a job shop, such as demanding scheduling and routing of multiple product variants. In particular, machines have frequent breakdowns and the system is constrained by bottleneck tools due to their high utilization. Although only 5 different processes are used to produce integrated circuits, these process

Thomas Altenmüller, Marvin Carl May, Andreas Kuhnle and Gisela Lanza authors contributed equally to this work.

✉ Sebastian Behrendt
sebastian.behrendt@kit.edu

Thomas Altenmüller
thomas.altenmueller@infineon.com

Marvin Carl May
marvin.may@kit.edu

Andreas Kuhnle
andreas.kuhnle@kit.edu

Gisela Lanza
gisela.lanza@kit.edu

[1] wbk Institute for Production Science, Karlsruhe Institute of Technology, Kaiserstraße 12, 76131 Karlsruhe, Germany

[2] Infineon Technologies AG, Am Campeon 1-15, 85579 Neubiberg, Germany

Published online: 29 January 2025

steps are repeated many times for individual products, creating recurring material flows. In addition, these processes have uneven process times (ranging from minutes to hours), making continuous material flow difficult. Finally, the processes are performed on different processing units (individual products, batches, or multiple batches). This results in a complex orchestration of batching and splitting of material flows. All of these characteristics of complex job shops contribute to the difficulty of operating semiconductor manufacturing systems efficiently.

One solution to overcome these difficulties and achieve operational excellence in semiconductor manufacturing is Industrie4.0 (or smart manufacturing), which exploits the information gain of digitization with sophisticated methods (Lasi et al., 2014; Waschneck et al., 2016). Digital twins (DT) in particular are one development of Industrie4.0 that promises to enable more efficient production through detailed analysis and forecasts (Lee et al., 2020; May et al., 2021). A DT aims, according to Grieves (2014), to be a detailed, indistinguishable virtual counterpart of a real physical system. Kritzinger et al. (2018) expand this definition by differentiating between digital models, digital shadows and DTs depending on the level of automated data flow. As stated by the authors, a DT has an automatic data flow from the real production to the digital system and vice versa. Thus, a DT automatically adapts to changes in the real production and can be used, more importantly, to analyze the production for potential problems, validate possible solutions and automatically deploy adjustments to resolve problems in the production (Kritzinger et al., 2018; Martínez et al., 2018). Since a digital twin should be used to generate or support decision, a DT allows for faster and better production planning and control by implementing foresight (Lee et al., 2020; Negri et al., 2017). Digital twins can be realized by a system of different kinds of models with interconnected information flow (Kritzinger et al., 2018). Since this paper focuses the analysis of production system performance, we will concentrate on material flow simulation models as virtual counterparts.

Current procedures for modeling and maintaining virtual representations of production systems are highly manual and, therefore, time-intensive tasks performed by domain experts (Lee et al., 2012; Martínez et al., 2018; Vernickel et al., 2020). In order to exploit the advantages of DTs, an automated procedure is needed that generates DT simulation models, as proposed by the concept of ASMG (Bergmann & Strassburger, 2010; Reinhardt et al., 2019). However, realizing such a procedure in a real production environment with the current data quality and availability of production environments is particularly challenging (Tliba et al., 2023). Production data in its raw form is typically noisy and may be incomplete, due to missing analysis for static, explicitly provided information, such as processing time distributions (May et al., 2023). Thus, the information required to parameterize a valid DT is mostly unavailable.

This paper presents a methodology for ASMG that resolves existing challenges and limitations by using data mining and machine learning for analysis of production data and a highly flexible material flow simulation framework that allows fully parameterized model generation, as illustrated in Fig. 1. The three key contributions of this research include:

- A fully automated, data-driven pipeline that extracts and filters real production logs (lot tracking, resource states) to generate simulation-ready parameters with minimal manual intervention.
- An integrated machine learning approach to accurately capture complex tool behaviors (e.g., batch, pipeline processing) and overcome noisy or incomplete data-critical for advanced semiconductor fabs.
- A generalizable validation framework that demonstrates our methodology's transferability and robustness across two distinct real-world semiconductor production environments, ensuring both throughput and resource utilization validity.

These contributions significantly reduce the time and expert knowledge needed to create and update comprehensive simulation models, paving the way for agile digital twin implementations in semiconductor manufacturing.

The remainder of this paper is organized as follows. Section "State-of-the-art" discusses recent approaches in the field of DT and automated simulation model generation to derive the key research question. By considering these research questions, section "Approach" introduces the methodology of this paper, consisting of a data mining pipeline and an automated simulation model generation procedure. The results of application and validation of the developed methodology in two real use cases are presented in section "Results". Section " Discussion" discusses gained insights and section "Conclusion" summarizes the findings and gives an outlook for future research.

State-of-the-art

In case of production systems, a DT can be realized by adaptive and highly detailed material flow simulation models (Negri et al., 2017). Digitization of production systems serves as an enabling technology for DT by creating large amounts of data that can be utilized by data mining techniques to model DTs (Farsi et al., 2020; Kritzinger et al., 2018). Challenges of realizing DTs with simulations in manufacturing are data gaps, lack of experts, high manual efforts in modeling and need for frequent adjustments (Kusiak, 2023; Oliveira et al., 2023). A solution to overcome these challenges are the

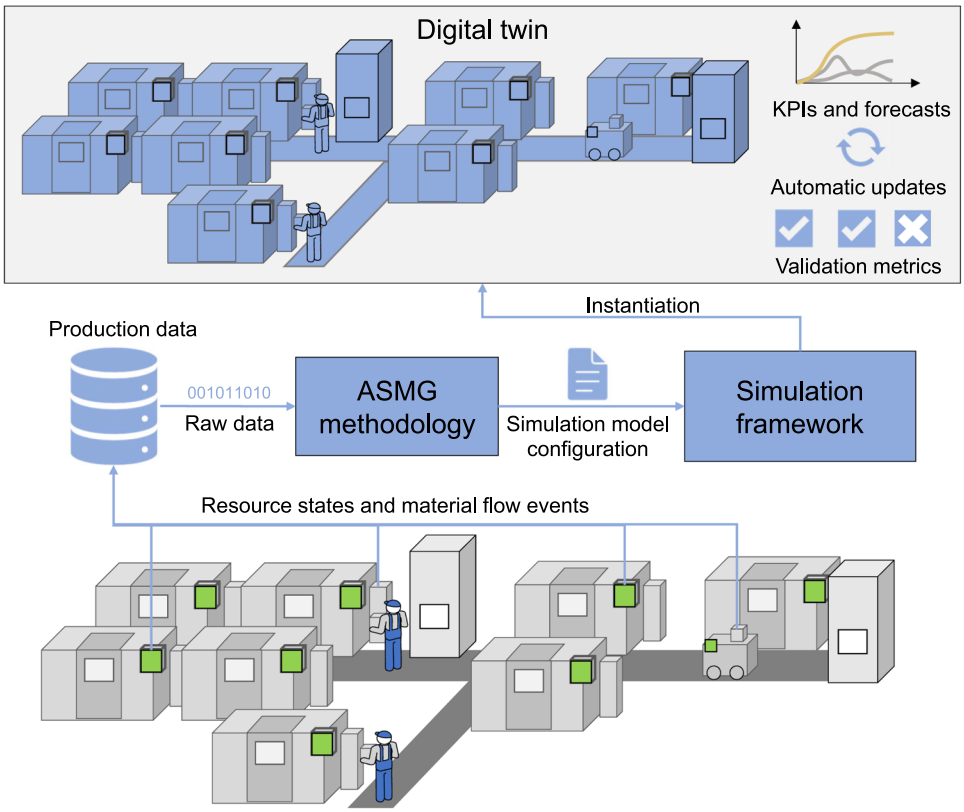

Fig. 1 Illustration of the research methodology for automated simulation model generation based on real production data

integration of data-driven modeling of components of a simulation model with machine learning (Camargo et al., 2020; Rathore et al., 2021; von Rueden et al., 2020). Numerous approaches in literature survey the potentials of schema-based data modeling for digital twins (Göppert et al., 2023). Yet, analyzing and aggregating real production data for these data models and dynamically representing the production system state with models, such as material flow simulations, needs to be researched more deeply (Choudhary et al., 2009; Overbeck et al., 2021). Several conceptual frameworks have been proposed for adaptive, data-driven simulation model generation, but modeling and updating of these models is in practice still a manual, time-intensive task for domain-experts (Reinhardt et al., 2019; Vernickel et al., 2020). One reason for this is that commercial-off-the-shelf simulation frameworks do not provide the required flexibility for a fully automated model generation from real data (Lee et al., 2012; Mourtzis et al., 2014; Rasheed et al., 2020). Moreover, there is a lack of examples in literature that realize the automated creation of a DT by the use of noisy production data (Milde & Reinhart, 2019), especially when considering a validation in the challenging environment of semiconductor fabs.

According to the concept of ASMG, a simulation model that resembles the real production should be created without any manual efforts by analysis of data from external data sources (Eckardt, 2002). Yet, existing approaches in the field of ASMG lack transferability to other use cases often caused by heterogeneous data sources and non-standardized data representations in enterprises (Reinhardt et al., 2019).

In the work of Milde and Reinhart (2019), for instance, an ASMG approach is proposed to automatically model and parametrize a manufacturing simulation model by manufacturing data analysis. However, the validation of the approach is only conducted on a idealized use case and data produced by another simulation. Bagchi et al. (2008) describe a procedure to automatically update simulation parameters for a semiconductor fab. The approach relies thereby on explicitly provided information, such as e.g. the process sequences, the current work in progress (WIP) and the tools in the production. Martínez et al. (2018) present an ASMG method for a digital twin of an industrial process plant. The data for the simulation is determined from a 3D plant model and an optimization method is successfully used to adapt model parameters to achieve higher validity. Transferability of the method to real production environments is limited due to need of manually maintained data concerning process equipment and applicability of the gradient-free optimization to small systems with few parameters. Lee et al. (2012) present an approach that transfers data from a product lifecycle management (PLM) system to a simulation package. An explanation how to consider dynamic production data, which is typically not available in PLM systems, is missing. A further approach that automatically models semiconductor equipment based on real production data is proposed by Kohn and

Werner (2010). The authors successfully model the dependency of processing time and lot size. A lot corresponds to a group of products that are transported together (Lee, 2008). Outliers in real production data prevent, according to the authors, reliable model generation. Therefore, model verification by experts is still necessary. Vernickel et al. (2020) show the possibility to include machine learning into simulation models by predicting process times with models trained on production data. An exemplary application of the concept on data created by a simulation model shows that the use of machine learning improves the validity of the simulation. Tliba et al. (2023) investigate the potentials of a DT for dynamic scheduling in a hybrid flow shop with utilization of a simulation for validation of robustness of the generated schedules. The simulation model is instantiated with static data from an enterprise resource planning system and dynamic data from the shop floor. Although the approach incorporates dynamic data it relies on provision of explicitly provided information from the ERP system, which may be not available or not up to date. Denno et al. (2018) present a methodology called production system identification which analyses production event logs with genetic programming and Petri nets. The approach mines only dynamic production data and incorporates probabilistic neural networks. Yet, explicitly provided causal models are needed to correctly model infrequent exceptional events that pose a serious problem in statistical analysis.

Reinhardt et al. (2019) encourage in their survey about the application of ASMG in manufacturing that future research should concentrate on approaches which can be universally applied and use dynamically captured production data. Bergmann and Strassburger (2010) support this statement by denoting the lack of universal validity and limited level of ASMG automation. Furthermore, the authors describe that another challenge of such approaches is the handling of missing but required information especially in the context of the dynamic behavior of the system. As proposed by von Rueden et al. (2020), machine learning techniques could be applied.

To resolve the current limitations of existing approaches for ASMG in regard of universal applicability and handling noisy or missing production data, three main research questions will be targeted:

1. What dynamically gathered production data can be used for ASMG in semiconductor manufacturing?
2. How does the data need to be filtered and analyzed to generate a material flow simulation model?
3. Is this automatic model generation transferable to different use cases in semiconductor manufacturing?

In the remainder of this paper, a methodology is presented and validated in real use cases with the aim of answering the posed research questions.

Approach

Figure 2 provides a step-by-step overview of our integrated ASMG methodology, emphasizing how real production data is transformed into simulation-ready inputs. The methodology is organized into five key phases:

- Data acquisition and Preparation (section "Data acquisition and preparation"): we query the relevant databases and perform outlier filtering to ensure clean inputs.
- Process modeling (section "Process modeling"): we derive arrival processes and material flow models by analyzing the historical trajectories of lots.
- Resource modeling (section "Resource modeling"): we classify tools (e.g., single, batch, pipeline) and capture their availability using breakdown and repair distributions.

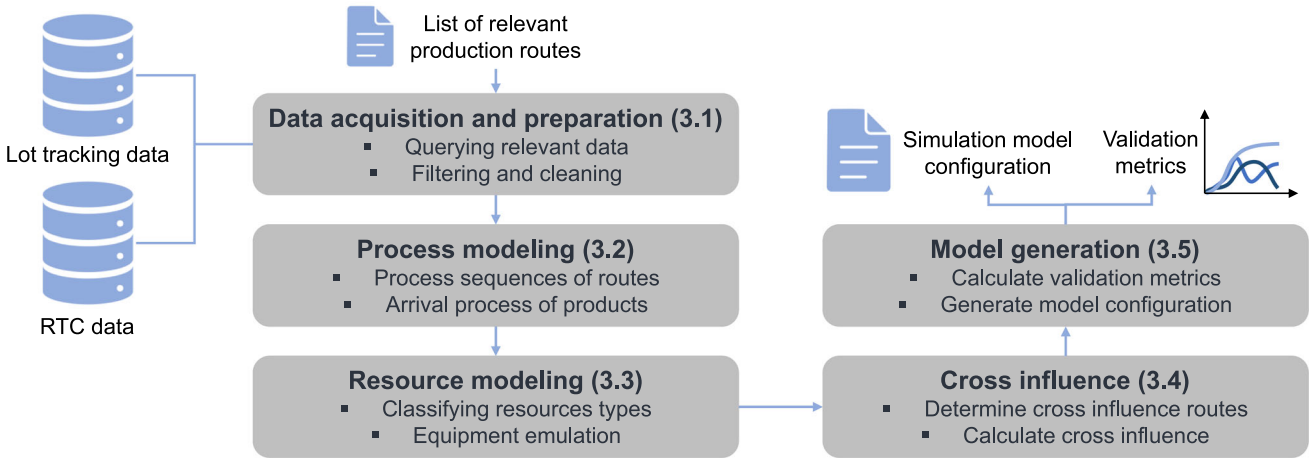

Fig. 2 Overview of the data mining pipeline to automatically create simulation models from real production data

- Cross influence (section "Cross influence"): we integrate the additional, non-modelled lots to reflect realistic workloads.
- Model extraction (section "Model extraction"): we compile all derived parameters and validate the resulting simulation against real production metrics.

Data acquisition and preparation

To address research question 1, we identified and evaluated data sources that serve as the foundational basis for ASMG in semiconductor manufacturing. The data sources were required to meet two key criteria: they must represent dynamic information reflecting the historical states of the production system and be readily accessible within typical semiconductor manufacturing enterprises.

Two event log data sources—lot tracking data and resource tracking and control (RTC) data—were identified as essential for ASMG due to their capacity to capture dynamic production events and resource states comprehensively. Lot tracking data captures detailed material flow information by recording each lot's transport and processing events, including start and end timestamps, associated resources, and descriptions of performed activities. An example of this data source is given in Table 1. A process refers in this context to a distinct production operation performed on a machine. Lot tracking data is automatically acquired through sensors and lot tags, which log events prior to each processing step or transport activity.

RTC data provides a chronological record of machine states, including events such as breakdowns, transitions from standby to production, and other status changes critical to understanding machine availability. An example for RTC data can be found in Table 2. Here, data acquisition is also automated by the controller of the resource which automatically sends information to data bases in case of a status change of the resource.

By relying exclusively on lot tracking and resource tracking and control (RTC) data, we avoid the dependence on manually curated data (e.g., process flow diagrams, maintenance logs), thereby improving scalability and timeliness. This exclusive use of automatically gathered event logs, a hallmark of most semiconductor fabs, underscores the method's broad applicability.

Considering the early digitization and automation in the semiconductor domain, automated acquisition of such material flow data is de-facto standard since the early 2000s of semiconductor manufacturing systems (Mönch et al., 2011). Although the semantics and structure might differ in companies, conceptually similar data sources are typically available in semiconductor manufacturing enterprises and are also heavily used in other data analytics, such as process mining. Therefore, the two data sources are well suited to answer research questions 1.

Starting point for the ASMG methodology is a list of production routes each specifying a defined process sequence that is used by multiple products. A process sequence specifies the sequence of processes required by products of a given route. Goal of the ASMG pipeline is to generate a simulation model that represents the production system used for manufacturing the specified production routes.

The design decision to use a list of routes as the input of the pipeline is based on the high number of production routes in semiconductor fabs. By selecting the material flows that should be explicitly modelled, the complexity of the generated models can be limited and the simulation tool can be used for specific purposes. This is especially useful as the occurrences of routes in a production system mostly follow a Pareto distribution, with a few high quantity routes and many low quantity routes (Bengtsson & Olhager, 2002).

Based on the provided routing information, databases are queried for the associated production data. At first, lot tracking data for a specified time frame, here 3 months, of the specified routes is retrieved. This lot tracking data is analyzed for the equipment used, and, in turn, the associated RTC data of this equipment is retrieved from the data bases.

Given the reliance on real production data, extensive preparation is required to mitigate quality issues, such as inconsistencies and imprecisions, thereby enhancing data reliability for modeling. Effective outlier filtering is essential to ensure that modeled distributions reflect normative behavior, enhancing the accuracy and reliability of simulation inputs. The method proposed by Tukey (1977) is used to determine and filter outliers, as this method is easy to apply and delivers robust results.

The results of filtered process time data and data modelled by normal distributions can be seen in Fig. 3a. Clearly, the data contains some outliers with processing times of more than 400 min. Applying Tukey's method with different values of k and fitting normal distributions on the data show that different amounts of outliers are kept in the modelled distributions. In our research, a value of $k = 2.5$ showed a good compromise of filtering extreme outliers and keeping some outliers. Retaining certain outliers is crucial, as they may represent process anomalies—such as delays caused by auxiliary resource shortages (e.g., reticles in lithography)—which are not explicitly captured in the lot tracking data. Therefore, these interruptions are considered implicitly be elongated process times. It is important to review distributions where data is strongly violated by outliers since this can indicate that a concept drift occurred. In these cases, updating lot tracking data or only considering more recent data points may be required.

In case of the RTC data, very short machine interruptions could distort the correspondingly modelled time distribu-

Table 1 Illustration of typical lot transaction data, including key attributes such as associated production equipment, a process identifier (SPS_NUMBER), and timestamps that allow to understand material flow events

LOT	ROUTE	EQUIPMENT	SPS_NUMBER	START_SPS	END_SPS
L0	R0	M0	5417	01.05.2023 14:47:21	01.05.2023 15:41:32
L1	R1	M4	8572	01.05.2023 15:22:29	01.05.2023 15:58:17
L2	R0	M7	5246	01.05.2023 15:45:57	01.05.2023 16:54:33

Table 2 Exemplary table of resource tracking and control (RTC) data that describes status changes of resources with specification of the associated equipment, the time stamp of the status change and type of status

EQUIPMENT	TIME_STAMP	PREV_STATUS	STATUS
M0	01.05.2021 14:47:21	SB	PR
M5	01.05.2021 14:53:34	SB	UD
M7	01.05.2021 14:54:33	SB	PR
M0	01.05.2021 15:43:17	PR	SB

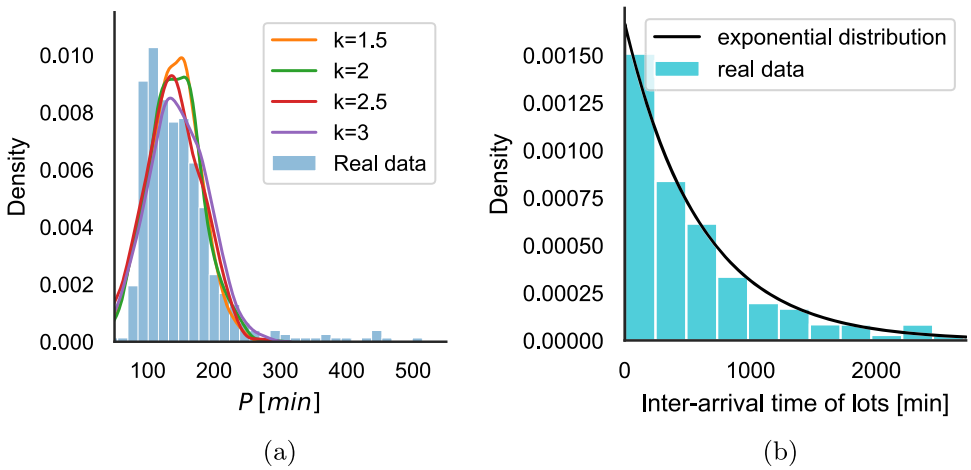

Fig. 3 Result of outlier filtered processing time data and fitted normal distributions (**a**) and inter-arrival time distribution of an exemplary arrival process and the corresponding modelled exponential distribution (**b**)

tions, as the most common approach relies on mean time between failure assumptions. Since such cases are typically artifacts of tests during maintenance, the interruption would distort the repair time and time between failures distributions. Thus, very short interruptions of machine states are filtered out. Another element in data preparation is to consider only traces that are used for production in the process modeling step. A trace refers here to the realized event sequence of an individual lots (van der Aalst, 2011). Lots that serve other purposes, such as development, rework or tests, are later implicitly considered by cross influences.

One critical data quality issue is the partial absence of transport data in the lot tracking datasets of the examined use cases, which hinders the direct calculation of transport times. To address this issue, interpolation techniques are applied to estimate values for incomplete data points, though this approach introduces potential inaccuracies and should be used with caution.

Process modeling

In the process modeling step, lot tracking data is analyzed for material flow properties, that are described for every individual route by an arrival process and a process sequence with associated tool dedications. An arrival process describes, with reference to queuing theory, the arrival of lots of a specified route (Dudin et al., 2020).

Process sequences are derived by analyzing the traces of all lots associated with specified routes in the lot tracking data, ensuring a comprehensive representation of historical material flows. Given the chronological sequence of performed processes and used resources, multiple traces are associated to a route. As the set of individual route traces exhibits only little variance and as concurrent events occur rarely, process discovery algorithms are not required to describe the most common process sequence. Instead, the traces of a route are combined to one process sequence, that contains all observed processes. As not all processes of a route are performed on every lot, e.g. due to lot sampling

at metrology processes, process probabilities are determined for each process in a process sequence.

Furthermore, tool dedications have to be discovered in order to describe the material flow completely. Tool dedications are defined as the relationship that certain processes are performed only by some tools (Shanthikumar et al., 2007). Thus, the set of used production resources is determined for every combination of process and route.

Lastly, the arrival process of lots in every route has to be modelled. Lot tracking data is used to calculate the inter-arrival time of lots in a route. The arrival process is modelled as a Poisson process with exponentially distributed inter-arrival times. The decision to use this particular model is motivated by its simplicity and the fact that it is a valid assumption for many real world systems (Dudin et al., 2020). Figure 3b shows the distribution of inter-arrival times of lots of an exemplary route and the modelled exponential distribution. Since the modelled distribution approximates the data well, the diagram suggests that the exponential distribution is well suited to model the arrival process.

As this methodology relies on historic data, the lot tracking data requires to be representative for the current state of the production system. Thus, timeliness of data is important. If a concept drift happens that is not represented yet in the data, associated information has to be explicitly provided to the simulation model. Possible reasons for concept drifts could be e.g. newly introduced routes or process or changed machine dedications. In the later use-cases, however, this was not required. Additionally, queue time constraints have also not been considered in the process modeling due to the infrequency of time constraint violations May et al. (2021).

In contrast to purely manual or partially automated approaches, our methodology systematically extracts routes, process sequences, and tool dedications by analyzing raw event logs. This dynamic extraction ensures the model stays current even as process flows evolve, unlike static modeling methods that rely on outdated documentation.

Resource modeling

The resource modeling procedure consists of a classification of the processing behavior (section "Extraction of processing types and process capacities"), equipment emulation to model state-dependant processing behavior (section "Equipment Emulation") and the modeling of the availability of production resources (section "Resource state modeling"). Additionally, machine groups are determined to reflect the job shop structure of the real production (section "Machine group determination").

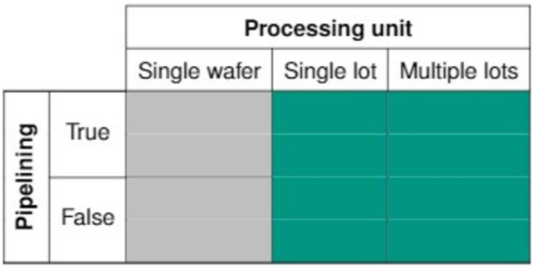

Fig. 4 Semiconductor processing types differentiated by their processing unit and whether they perform pipeline processing

Extraction of processing types and process capacities

The proposed classification scheme systematically identifies and parameterizes the distinct processing behaviors of semiconductor production resources, accounting for processing unit types and pipeline capabilities. Figure 4 shows this scheme that classifies the processing types based on the processing unit and whether the processes are performed in a pipelining manner. According to Mönch et al. (2012), the processing unit of semiconductor production resources can be a single wafer, a lot or a number of lots. A wafer is a flat silicon disk which is the smallest processing unit in fronted semiconductor manufacturing. Typically, these wafers are grouped in lots of 25. Although single wafer processes are performed sequentially on individual wafers, the wafers of a lot are generally processed on the same machine since wafers are transported together in lots (Lee, 2008). Processes can either be performed strictly sequentially or in an overlapping manner, i.e. pipelining, for consecutive processing units of a manufacturing tool. Pipelining is mainly the case for cluster tools, a special case of manufacturing equipment where multiple processing chambers can individually execute processes on the processing unit. Moreover, many processes are also performed as batch processes on multiple lots, e.g. heating. Both processing unit and whether the tool performs pipelining have a major impact on the processing behavior of the tool and have therefore be considered in data analysis.

Since we take a strictly empirical approach based on lot tracking and RTC data only, no information is available about the processing of the individual wafers of a lot. Therefore, classifying processes of resources based on lot tracking data as single wafer processes is not possible. To overcome this problem, tools with single wafer processes, such as physical vapour deposition tools, have to be treated as black box models, where only the start and end of processed lots are considered. The internal processing of single wafers, however, is not modelled but statistically considered. This simplifications motivates the later use of machine learning to enhance solely statistical models by learning the processing behavior from historical data. Nevertheless, it has to be determined which of the green colored processing types in Fig. 4

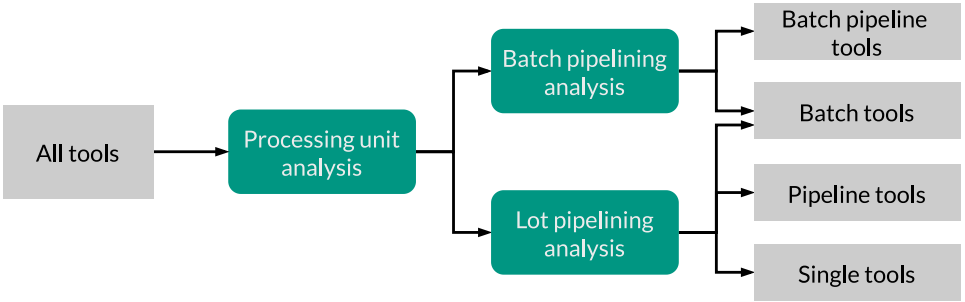

Fig. 5 Approach to separate the processing unit types based on a two stage analysis considering production unit and whether pipeline processing is performed

suits the processing behavior of the individual production resources best.

Figure 5 illustrates the procedure to solve this classification problem based on the processing behavior of tools observable in lot tracking data. First, the tools are divided into two groups based on their processing unit. As a second step, the pipelining behavior of the tools in the two groups is analyzed independently and a processing type is obtained.

A tool's processing unit type can be derived from its batch size distribution present in lot tracking data. A batch size B corresponds to the number of lots that are simultaneously processed (Shanthikumar et al., 2007). With the condition that the processes of lots started and ended respectively within a range of 1 min, the processed lots of a tool are grouped into batches. The classification of the processing unit of the tool is then performed by comparing the mean batch $\overline{B}$ size to a threshold of $\overline{B} = 1.25$. Only if a tool's mean batch size exceeds this threshold, its processing unit is classified to multiple lots. We selected a threshold larger than 1, in order to avoid faulty classifications due to data quality problems.

Tools that are not classified as batch tools are analyzed for their lot processing behavior. To analyze if pipelining occurs, a measure is needed that allows to assess how concurrently lots are processed at these tool. When considering a processing sequence at a tool as depicted in Fig. 6b, a measure for the concurrency of the process of lot L0 could either be based on counting overlapping processes or it could be based on the overlapping process times, which are displayed by the scattered parts of the process time.

The measure C_d, called discrete concurrency and defined in this research, follows the first approach and is defined as the number of lots that are processed on a machine at the time of a newly started process. In the example, the process of lot L0 has a discrete concurrency of $C_d = 3$ given that the machine is processing lots L1 and L2 at the processing start of lot L0. It should be noted that the process for which the discrete concurrency is calculated, is also considered. Thus, discrete concurrency is limited to a minimum of $C_d = 1$.

The measure C_c, called continuous concurrency, is motivated by the second approach, namely describing the concurrency of a process based on the overlapping process times. The formula to calculate the continuous concurrency for the process of a lot is given in Eq. (1). In the formula P refers to the process time of the process for which the measure is calculated and $\sum P_o$ is the summed overlapping process time of processes from other lots. For example, the process of lot L0 from Fig. 6c has a continuous concurrency of $C_c = 2.5$.

$$C_c = 1 + \frac{\sum P_o}{P} \tag{1}$$

In contrast to the discrete concurrency, the continuous concurrency is a floating point value, and can be used to measure at what degree the processes are performed in parallel. If the continuous concurrency of a process is an integer (unequal to 1), it means that the process is performed most likely in a batching manner.

The classification as a pipelining tool is based on the idea to evaluate if the tool's processing behavior is more similar to that of a batch tool or to that of a pipelining tool. One possibility to create a feature, called continuous concurrency deviation ΔC_c, that characterizes the processing behavior is shown in Eq. (2). The feature calculates the average distance of the continuous concurrency value to the closest integer value for every lot l from the set of all lots L that are processed on a tool.

If the continuous concurrency deviation is close to 0, it means that the continuous concurrency is close to an integer number for the most values, which resembles the processing behavior of batch tools. A higher value of this feature means that the continuous concurrency values for the processes of the lots are further away from integer numbers, which reflects the processing behavior of a pipelining tool.

$$\Delta C_c = \frac{1}{L} \sum_{l=1}^{L} |C_{c,l} - \lfloor C_{c,l} \rceil| \tag{2}$$

Another feature used in the lot pipelining analysis describes how strongly the process time depends on the continuous concurrency. This is motivated by the fact that for non batch tools the process time will increase when more lots are processed in parallel. This effect is caused by the way the processing time is logged in lot tracking data. The time stamp for the start of a process does not necessarily reflect the physical

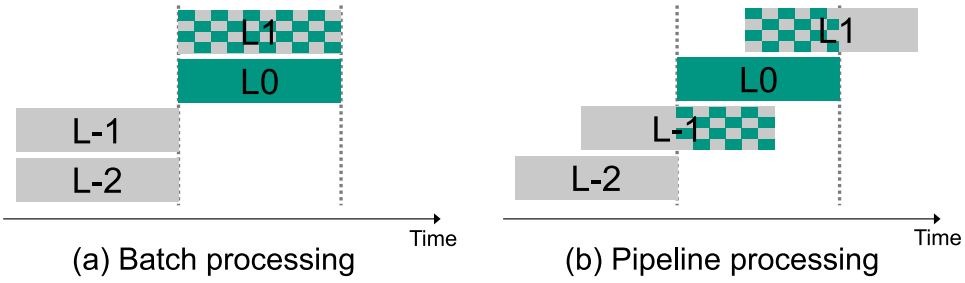

Fig. 6 Visualizing the difference of batch processing (**a**) and pipeline processing (**b**) with consideration of the overlapping processing time in the checked squares

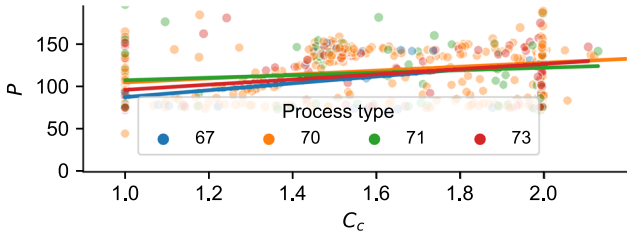

Fig. 7 Visualization of the dependency of processing time and continuous concurrency for an exemplary processing tool over different process types

process start, but instead when a lot is loaded in the load port of a machine. In case of a machine with multiple load ports, a lot can wait in the load port while another lot is processed in the process chamber. Therefore, the process time increases with a higher number of concurrently processed lots.

Figure 7 illustrates this behavior as the processing time increases for a higher continuous concurrency value for all processes of this tool. The accumulation of processing times at $C_c = 1$ and $C_c = 2$ indicates that a significant percentage of processes are performed sequentially or in batches of 2.

The measure for the dependency of process time and concurrency is the slope of a uni-variate linear regression that predicts process times based on the continuous concurrency. A high value of this feature implies that the process time increases with higher continuous concurrency values and vice versa.

Based on continuous concurrency deviation and the slope of the process time, a clustering algorithm divides the processing equipment into two groups. The tools in the group with higher values of the two features are classified as pipeline tools, whilst the others are classified as batch or single tools, depending on their capacity. The clustering uses the K-Means algorithm, where $K = 2$ is specified with the aim of finding two clusters. The values of the two features are standardized for clustering due to the sensitivity of K-Means clustering to differently scaled features.

The result of the processing type clustering can be seen in Fig. 8, where every data point in the graph reflects the historic processing behavior of one processing equipment. The blue cluster has the lower values of the two features, which suggests that this cluster contains processing equipment which has a processing behavior more similar to batching tools. In contrast to that, the yellow cluster contains the processing equipment with higher feature value. Thus, this cluster comprises tools that are more similar to pipelining tools. The equipment type information, which is obtained from a tool overview list, supports this hypothesis as all cluster tools, which are able to perform pipelining, are in the yellow cluster.

The question might arise why single tools are classified as pipelining tools since they possess only one processing chamber and should not be able to perform overlapping processes. This is due to the logging of lot tracking data where time intervals between start and end of a process additionally include loading and handling times. Thus, a single tool with multiple load ports can start loading a lot while another lot is still processed, resulting in an observed pipelining behavior.

According to the data, most processing equipment process multiple lots in parallel. Thus, the equipment capacity, i. e. the maximum number of lots that can be processed at once, has to be determined accordingly. The discrete concurrency can be used to evaluate the equipment capacity because it describes how many lots are in the machine at the start of a process. Using an aggregation function on the distribution of discrete concurrency values of a tool, the equipment capacity can be determined. A good compromise uses the third quartile (Q3 aggregation function) of the distribution to determine the capacity, as it results in a neither too optimistic nor too conservative capacity assumptions.

Similar to the procedure of the lot pipelining analysis, it has to be checked if pipelining occurs at tools that have batches as processing units. However, for these tools it is only necessary to determine their equipment capacity by the number of batches that can be processed in parallel. Similar to the pipelining analysis, the discrete concurrency distribution, here for batches, is aggregated with the Q3 aggregation function to determine the equipment capacity.

The previously mentioned dependency of processing time and continuous concurrency occurs on all previously classified single and lot pipeline tools. Because a simple processing time distribution is not able to reflect this dependency, another modeling procedure is needed to consider this behavior in the simulation model.

Equipment emulation

A key advance of this work is the equipment emulation: we employ Random Forest and Extreme Gradient Boosting to learn intricate concurrency-dependent process times, bypassing the need for explicit domain knowledge (e.g., sub-chamber states) or additional sensors. This black-box approach accommodates incomplete or noisy data, yet consistently outperforms simpler statistical models in capturing real-world tool dynamics.

Equipment emulation is a supervised learning problem, more specifically a regression problem, that utilizes production data to learn the relationship of process time of a tool and the state of the processing equipment. A model is trained to approximate a function that describes the relationship of the target variable, i.e. the process time, to other features, which can be observed in Table 3 and Fig. 9. A data set for the machine learning can be obtained solely from lot tracking data since it contains the target variable and all features can be calculated from it.

The process time in cluster tools or single tools with multiple load ports is increased if multiple lots are in the tool since waiting time can occur then. The features used in equipment emulation for predicting the process time consider these effects and aim to give a quantification the processing state of the tool.

A dataset used for training, validation and testing is created for every selected tool. The dataset contains for every processed lot at the tool information about the process time and the feature values, according to Table 3. These datasets are then used by machine learning algorithms to train equipment emulation models for all pipeline tools. The ensemble methods Random forests and extreme gradient boosting (XGB) (Chen & Guestrin, 2016), an extension of gradient boost trees (Friedman, 2001), showed a good performance. Since dozens of trainings have to be performed to create the equipment emulation models for multiple tools, we used the same hyper parameter setting for Random forests and XGB with maximum 100 decision trees in every ensemble and a maximum tree depth of 5. This strategy held tuning efforts low by using the ability of Random forests and XGB to hardly overfit to data (Chen & Guestrin, 2016; Friedman, 2001). Transferring this strategy to neural network based approaches, such as LSTM or CNN, was not promising since the varying length of data sets of different tools from a couple of docent up to multiple thousand data points made hyper parameter tuning necessary. Therefore, only Random Forests and XGB are evaluated.

Figure 10 displays the performance of trained random forests and XGB models in comparison to linear regressions in a fivefold cross validation for all relevant tools in two semiconductor manufacturing production systems, that will be explained later in more detail. The diagrams show that for most modelled tools a R^2 value in range of 0.5 can be obtained. Random forests shows in both use cases a slightly better performance. The low values of R^2 can be explained by the stochastic nature of the processing time, making it hard, if not impossible, to predict correctly. However, a prediction that fits in average is for equipment emulation good enough.

Resource state modeling

Setup, engineering, maintenance and breakdowns of equipment reduce the overall time when the equipment is available to manufacture. These times have to be included in a simulation model to represent the uptime behavior of tools realistically. Information to calculate such time distributions is derived from the RTC data.

For a breakdown, the the time to repair (TTR) is calculated by the timely difference of start and end of a unscheduled downtime (UD) event. Moreover, the time to failure (TTF)

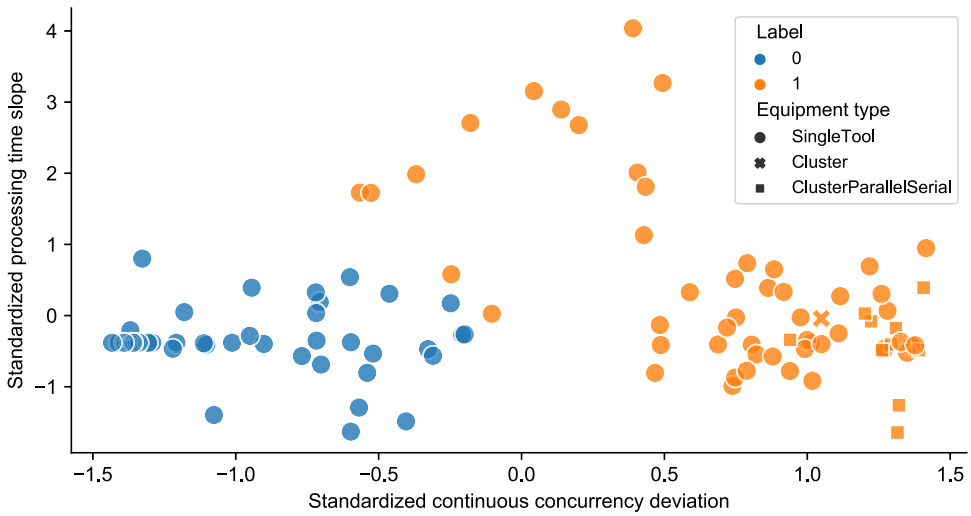

Fig. 8 Visualization of the result of K-Means clustering of processing equipment based on the standardized continuous concurrency deviation and the processing time slope

Table 3 Description of the features used for equipment emulation

Feature	Definition
Process Type	One-hot encoded SPS_NUMBER of the process
QTY	Number of wafers in the lot
C_d	Discrete concurrency, see section "Resource modeling"
Inter-start time	Time since last process start
P_{-1}	Process time of the last stated process
Sum finished process time	Sum of the finished process times of lots that are currently processed in the machine
Sum overlap process time	Sum of the overlapping process time of the lots that are currently processed in the machine
Finished ratio	Ratio of the sum finished process time and the process time of lots that are currently processed in the machine
Remaining ratio	Ratio of the sum overlap process time and the process time of lots that are currently processed in the machine

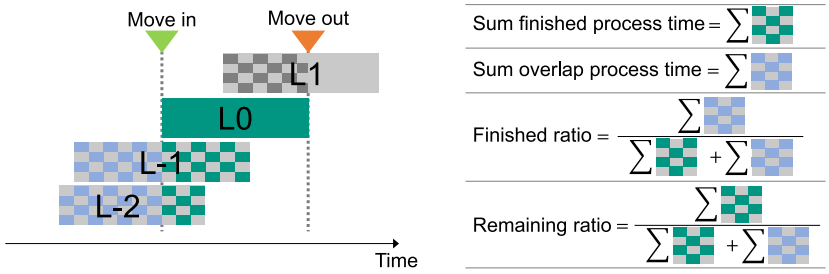

Fig. 9 Visualization of the calculation of the concurrency dependant features used for equipment emulation

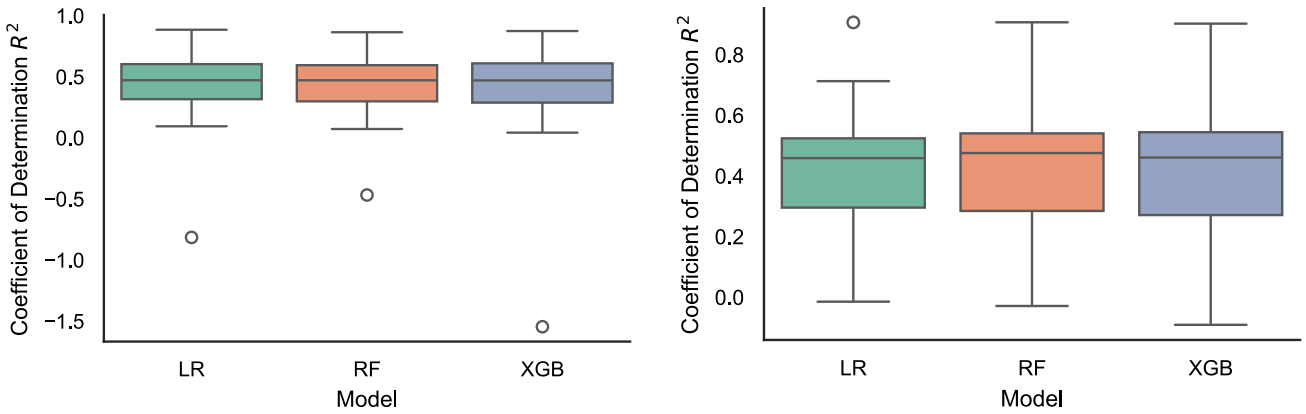

Fig. 10 Performance distribution of linear regression (LR), random forest (RF) and extreme gradient boosting (XGB) in a fivefold cross validation for equipment emulation of relevant tools in use case one (left) and two (right)

can be calculated by the time difference between the end of an UD event and the start of the next UD event. Similarly, time measures for lengths of events and time between events can be calculated for other event types. The distributions of time of events and time between events are then modelled by a log-normal distribution, as the actual data is well described by these distributions (Keller et al., 1982). Although occurrence and length of setup or scheduled downtime might be in most companies not completely randomly distributed due to planning, an assumptions is made that these effects are negligible considering length of simulation runs and size of simulation models.

In summary, all the tools present in the lot tracking data are modelled by this procedure and are assigned to a process type that describes the processing behavior. In order to parameterize the tool, the capacity is determined and the availability of the tool is modelled. For pipeline tools, equipment emulation models are trained that model the processing behavior more accurately than solely statistical models.

Machine group determination

As semiconductor manufacturing production systems are job shops, machines are organized in groups where all machines in a group can perform the same processes. Therefore, the

last step to model the production system is to determine the machine groups of the tools.

The previously obtained tool dedications specify for every process the set of possible tools. These sets of tools are arranged to sets of tools that have no redundancy and no intersections since machines need to be assigned to one particular machine group. Firstly, redundant sets of tool dedications are removed, resulting in unique tool dedication sets. Secondly, the tool dedications are analyzed for subsets, which are removed. Thirdly, it is checked whether remaining sets have intersecting tools. If this is the case, intersecting sets are combined. This results in the sets that describe the machine groups, which can be used in the simulation to organize the tools in groups.

Cross influence

The material flows modelled in section "Process modeling" exclude certain processes observed in the lot tracking data, necessitating the inclusion of them to account for their dynamic interactions and a representative system load. In order to resolve this problem, not explicitly modelled material flows are considered in the simulation by a concept called cross influence.

The complex job shops of semiconductor manufacturing and their intertwined material flows make it difficult to evaluate individual areas of the production independently. Cross influence resolves this problem by including the static influence of non-modelled material flows in the regarded system. Examples for non-modelled material flows are rework and engineering lots or lots of routes that only use individual tools of the modelled production system. With this, all machines in the regarded system have a realistic inventory and utilization but we can limit the explicitly modelled complexity of the simulation.

For cross influence, we assume that lot arrival times of the non-modelled lots, i.e. the cross influence lots, at modelled machines are not correlated, because these lots are entering the modelled production line from outside of the modelled equipment set, run a single process step on one of the modelled equipment, and leave the line again. This assumption allows to separate the processes of the cross influence material flows and model them independently. Thus, cross influence lots are modelled as a one step, dummy process and therefore only increase the workload of tools statically.

Figure 11 visualizes the concept of cross influence material flows. From the graph we can see that cross influence lots perform only one process step on a certain machine group, contrarily to modelled routes that perform a sequence of processes on multiple machine groups.

Cross influence material flows are determined for every process that occurs on a machine group and their arrival process is modelled as described in section "Process modeling". Actual production data shows that the inter-arrival times fit well to an exponential distribution.

Cross influence is vital for capturing realistic shop-floor interactions where engineering or rework lots add dynamic loading. Unlike older methods that exclude these lesser flows or rely on average overhead factors, our approach enables accurate simulation of total workload by automatically generating single-step processes for cross influence lots. This preserves system realism while maintaining manageable model complexity

Model extraction and validation

The final phase of the data mining pipeline involves extracting a comprehensive simulation model and rigorously validating its alignment with real production KPIs. The information that describes the model has to be compressed in a format that can be used in the simulation framework as an input. In order to perform the validation, validation data is calculated from the production data and simulation data by aggregating them to two comparable sets of KPIs. Comparing the resulting KPI distributions allows to assess the validity of the model.

The simulation model can be extracted by compressing the information obtained in the modeling section. The structure of the production system is described by the mapping of tools to machine groups. Each tool is characterized by its processing type, capacity and the parameters to model its process time distribution and machine state distribution. The material flows are described by the sequence of processes, tool dedications of these processes and parameters to model the arrival process distribution. Similarly, information for every cross influence process on every machine group is given by the parameters of the arrival process and process time distribution.

The KPIs to validate the simulation model are selected based on the most established KPIs in literature: throughput, uptime utilization, availability and cycle time. All the KPI values are obtained from lot tracking and RTC data. In order to describe these KPIs by distributions, the KPIs are calculated individually for every day. Throughput and cycle time are calculated for every route individually, and availability and uptime utilization are captured for every machine group. The simulation results are aggregated similarly.

Results

The following section presents the results of applying the methodology on two real semiconductor manufacturing production systems. The use cases and obtained simulation models are briefly described, and validity of simulation mod-

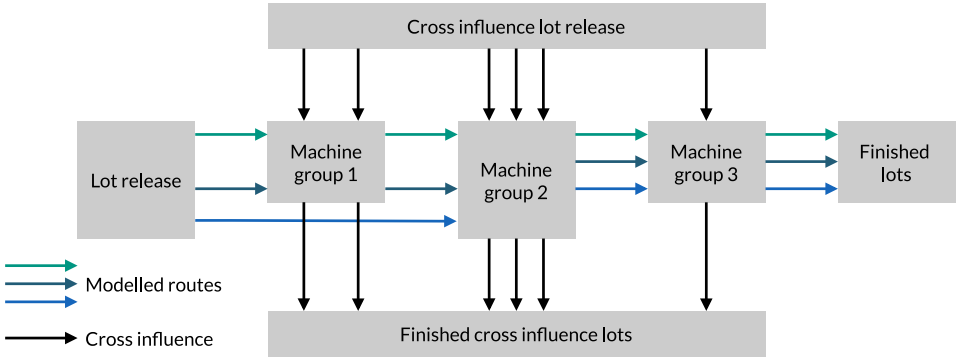

Fig. 11 Schematic illustration of the integrated cross influence concept

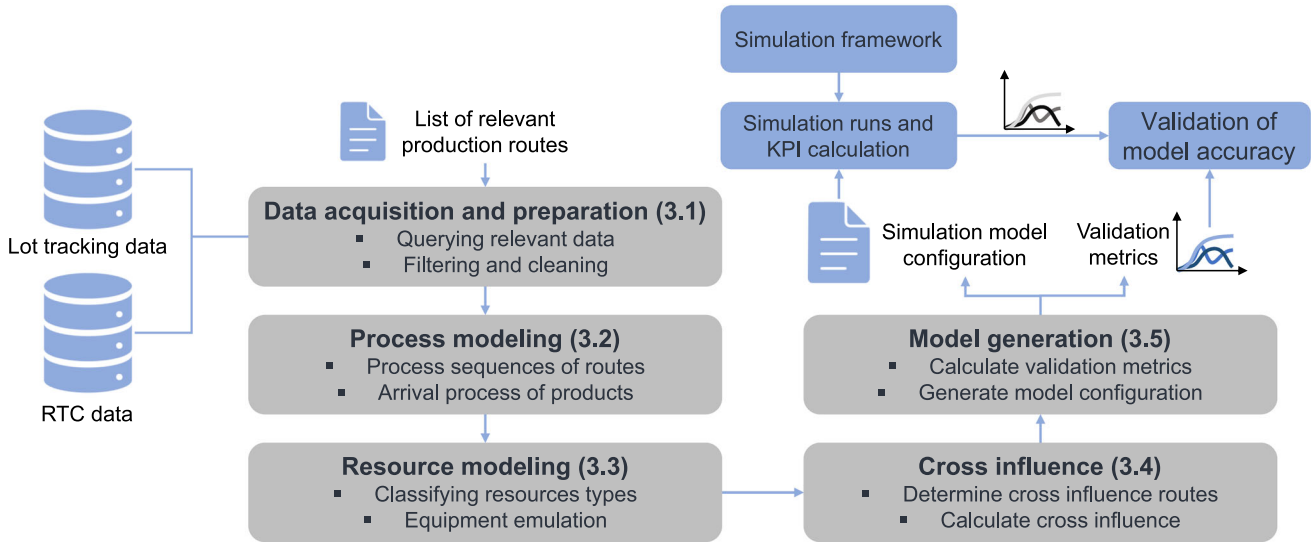

Fig. 12 Overview of the ASMG methodology to automatically create simulation models from real production data with the procedure for validating generated simulation models

els is assessed by comparison to the performance of the real production systems as visualized in Fig. 12.

The used production data for both use cases covers a time period of two months for lot tracking data and six months for RTC data. A longer time period is considered for RTC data, as breakdown events occur quite infrequently and the statistic over two months is too volatile. Validation of the obtained simulation models is done by assessing the similarity of KPIs of simulation model and the real production system. The associated simulation runs are performed for a time period of 300 days as this time frame ensures statistically reliable results. In order to make the simulation results reproducible, random seeds are used for stochastic variables. The warm-up period of the simulation run is discarded according to the MSER-5 method proposed by White (1997).

The two generated production system simulation models of the use cases are summarized in Table 4. When comparing them, it can be seen that the production system of the first use case contains less individual tools, but the number of machine groups is higher. The production system of the second use case consists of larger machine groups and has therefore more capacity for parallel processing. Another difference is the distinct composition of the tool's processing types in the two use cases. The first use case has a large number of single tools (ST), while the other use case has a higher percentage of pipeline (PT), batch (BT) and pipeline batch tools (PBT).

Table 4 Summary of equipment characteristics in the two analyzed use cases, categorized by the number of tools, machine groups, and processing types for comparison

Use case	Equipment	Machine groups	ST	PT	BT	PBT
1	83	43	35	36	9	3
2	112	32	9	52	38	13

The first KPI that will be examined for validation is the throughput, as shown in Fig. 13. In the diagram, the throughput over simulation time is displayed in lots per day for an exemplary route of the second use case. The exemplary route is representative for other routes of this use case. The real throughput of the production system is displayed by its average with a gray line, Q1 and Q3 of the throughput distribution

with the dashed gray lines. The shaded area between Q1 and Q3 symbols the range of the throughput of the route in the real production over the regarded time frame. As can be seen, the simulation model results in a throughput that is close to the real average value and in the range of the real throughput. The first use case yields the same observations.

The implementation of machine state distributions can be evaluated by reviewing the uptime distribution with respect to the real distributions.

Figure 14 shows the uptime distributions of real production and simulation for each machine group (MG) in the second use case. The uptime of some machine groups, such as MG8, is a constant value of 1 which suggests that RTC data is not available for these machine groups. MG31 is an example where some data is available, as the real data of this machine group suggests. But there is too few data available to model a distribution which is why the resource is assumed to have no breakdowns and the simulated uptime is always 1. Moreover, it can be seen that the real uptime distribution, as e.g. MG13, has in many cases a higher variance than the simulated distribution. An explanation for the smaller variance in the simulation is the use of outlier filtering and the fact that the log-normal distribution is used to model the machine state times. However, the modelled uptime behavior is for most machine groups in range of the real production. Figure 15 displays the mean value of the simulation distribution normalized with the real mean for all machine groups, showing that differences only range up to a maximum of 7.5% while mostly being less. Again, the first use case shows similar results with respect to the uptime distributions of machine groups.

Both throughput and uptime distributions are a direct result of simulation input parameters and are therefore expected to be valid. The dynamical validity of the simulation can be assessed more precisely by reviewing inventory level, uptime utilization and cycle time. This is based on the fact, that the modeling of production resources strongly influences these parameters. Thus, the correct modeling of these resources needs to be evaluated.

The uptime utilization distributions of real production and simulation are displayed in Fig. 16 for each machine group in the first use case. Most machine groups have an uptime utilization in the simulation that is close to the real distribution. This suggests that the tools of these machine groups are modelled correctly. Figure 17 underlines this statement, as it show that for most machine groups the mean uptime utilization of machine groups in the simulation normalized with their real uptime utilization are close to 1. However, it can be seen that some machine groups strongly deviate in simulation and real production, such as MG0 or MG12. In the investigated use cases, strong deviations were mostly present for machine groups with special tools, such as metrology tools or loader tools. This indicates that special treatment in the modeling of such tools is necessary. The uptime utilization distributions of the machine groups in the second use case allow for a similar observation.

Figure 18 illustrates the inventory level of the production system of the first use case. The real inventory level over the time frame of the lot tracking data is displayed in black. The diagram compares the resulting inventory levels of the simulation parameterized for different numbers of transport resources. Since utilization of the machine groups is mostly valid, a possible explanation for discrepancies between real and simulated inventory level could be due to inefficient material transport between the tools. Since transport times are also mostly missing in the raw data, transport could be unrealistically modelled in the simulation.

With the aim of conforming this hypothesis, a set of simulation experiments is conducted with different numbers of available transport resources. The results show that a limited number of transport resources leads to longer waiting times for transport and a higher inventory level. Without a restricted transport capacity, the inventory level in the simulation is significantly lower than that of the real production. This shows that inventory levels are highly sensitive to assumptions regarding the available transport capacity, underscoring the importance of accurate transport modeling. By determining a suitable level of transport restrictions, as displayed by the green line, a valid inventory level can be obtained, even in the absence of accurate raw data for transport.

When reviewing the cycle time of machine groups in the first use case, a similar observation can be made, as expected due to Little's law (Little, 1961). The cycle time of a machine group refers to the time passed between finishing a process at a machine group and the point in time when the previous process of the lot was finished. The associated distributions of real production and simulation are depicted in Fig. 19. Again, two simulation results are shown, one for an unlimited number of transport resources in orange and the other for a limited number of transport resources in green. It is visible, that the real production has for some machine groups, such as MG17 or MG19, much higher cycle time values that the simulated ones. As the uptime utilization of these machine groups is validly represented in the simulation, as can be seen in Fig. 16, it is likely that the increased cycle time is caused by inefficient operation of these machine groups and therefore increased waiting times. Although limiting the number of transport resources allows to increase the validity of the inventory level, the limitations of this methodology can be seen here. The inventory level and therefore the cycle times are increased the most at specific machine groups, as e.g. MG7. This, however, does not reflect the real inventory level distribution in the production system. Still, the limitation of transport resources gives insights about possible operating problems in the production systems and hints how to address improvements in the validity of the simulation. The same

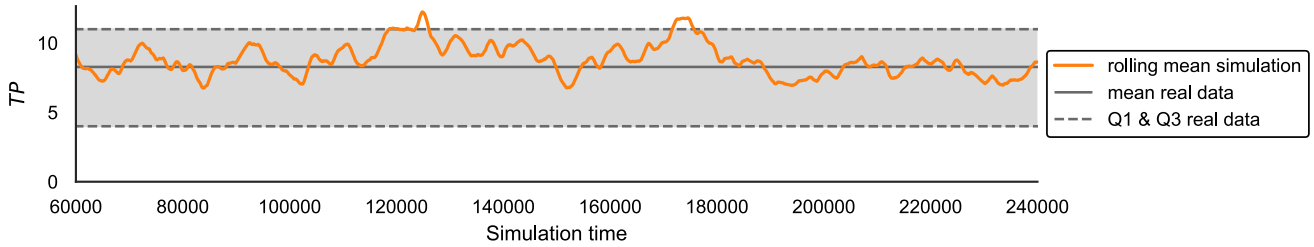

Fig. 13 Validation of throughput of simulation model with real production throughput for an exemplary route of the second use case

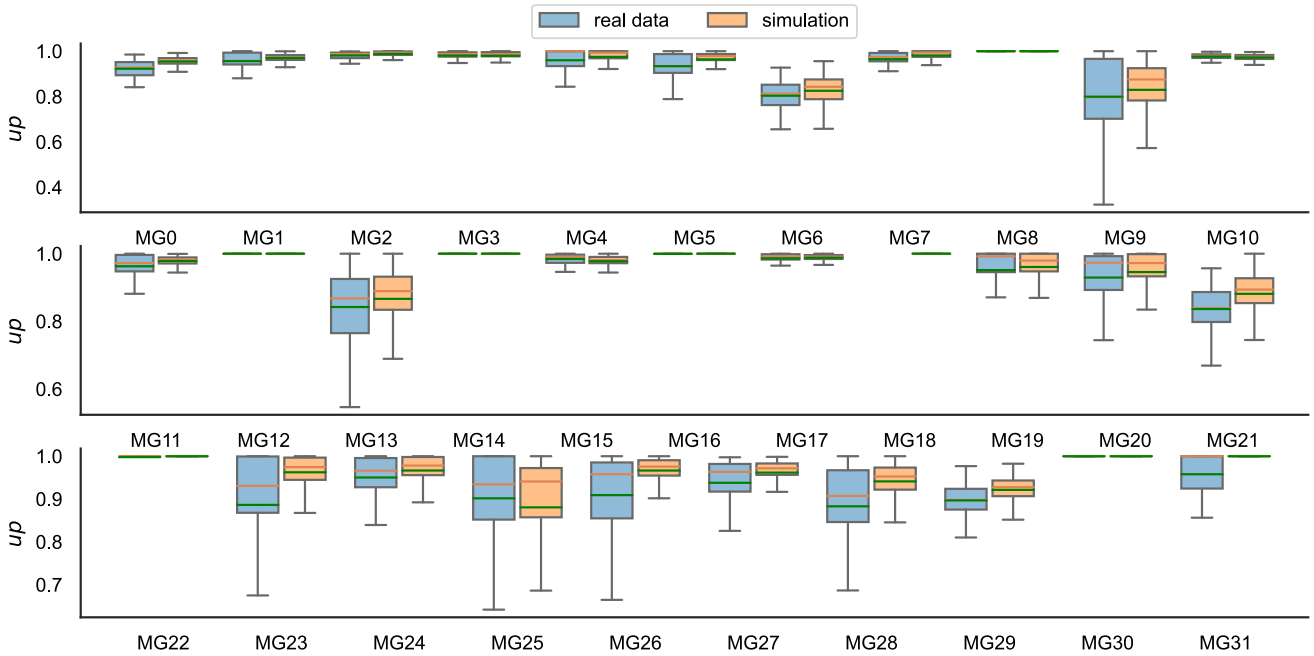

Fig. 14 Validation of uptime distributions of machine groups of simulation model with real production uptime of the second use case

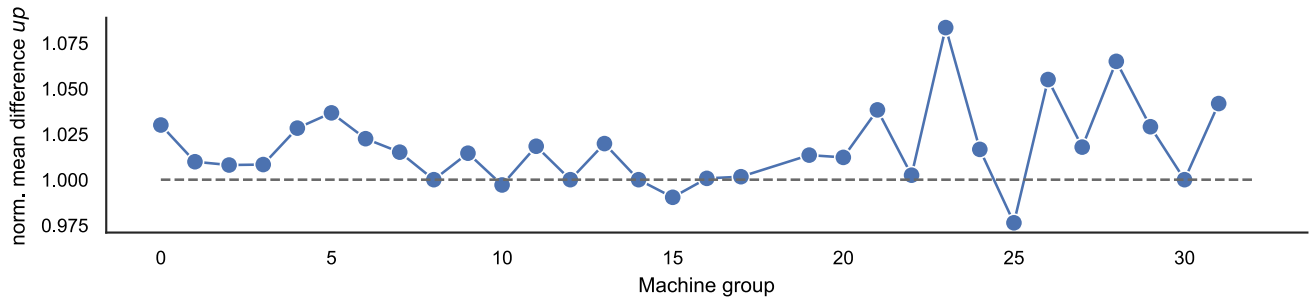

Fig. 15 Normalized mean of uptime distributions of machine groups of simulation model with real production uptime of the second use case

observations and insights can be made by analysis of the results of the second use case.

Discussion

In this paper, we address three main research questions through:

- Leveraging standard event log sources (lot tracking, RTC) to automatically retrieve dynamic data.
- Implementing a pipeline that uses data mining and machine learning to derive route information, tool behavior, and cross influence streams.
- Validating the universal applicability on two real-world production systems, highlighting throughput and utilization accuracy.

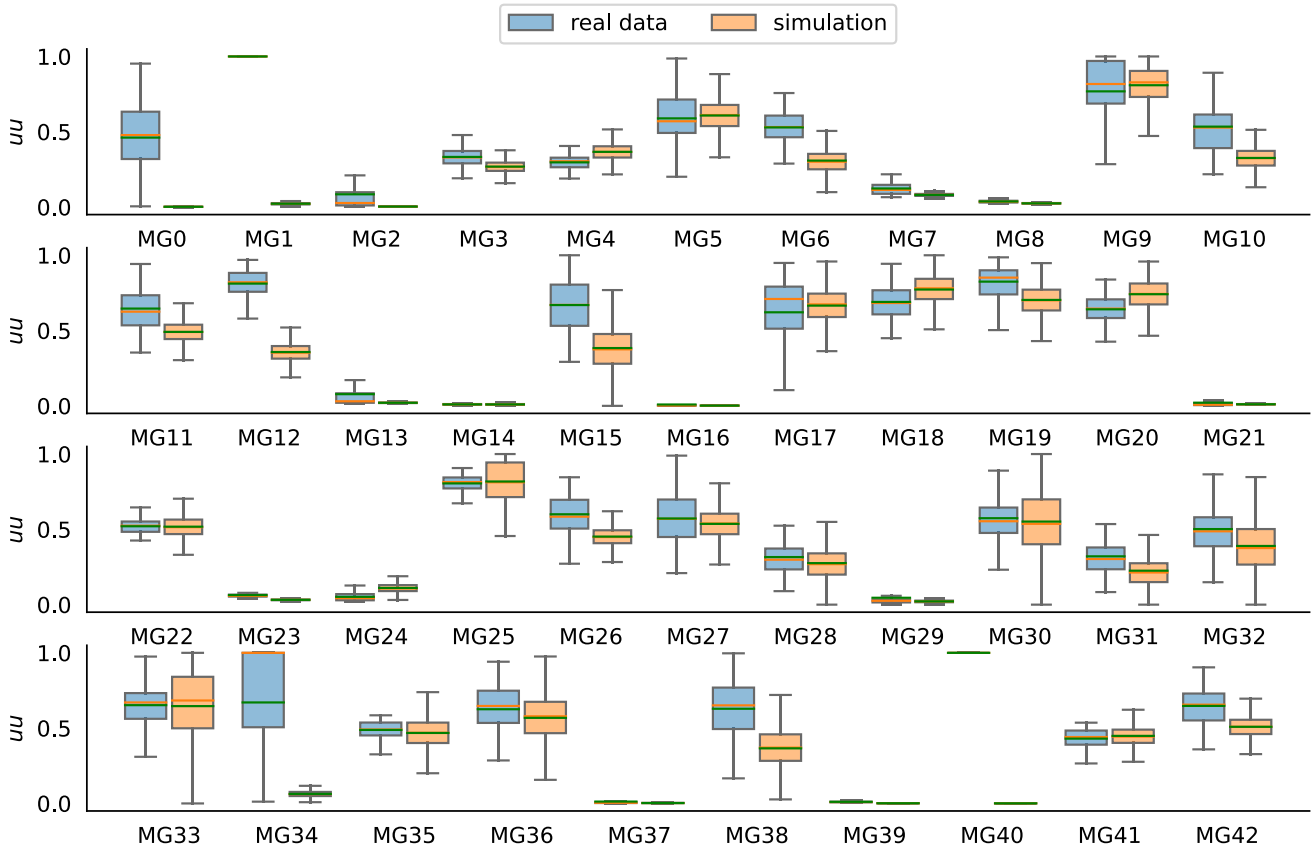

Fig. 16 Validation of uptime utilization distributions of machine groups of simulation model with real production uptime of use case one

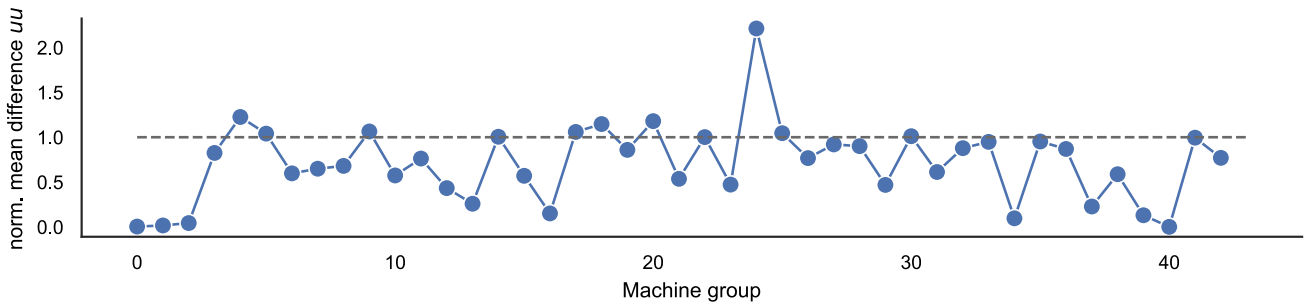

Fig. 17 Normalized mean of uptime utilization distributions of machine groups of simulation model with real production uptime of the second use case

These contributions demonstrate the potential of a fully data-driven, automated simulation model generation (ASMG) approach and pave the way for next-generation digital twins in complex semiconductor fabs.

The results show that the methodology enables to automatically generated valid simulation models without requiring any manual modeling efforts. No manually maintained data is required as the methodology solely relies on automatically gathered production data that is available in most semiconductor manufacturing environments. This also shows the transferability to other use cases without any manual adjustments. Limitations of data accuracy and missing information are overcome by utilizing machine learning in equipment emulation, resulting also in a higher accuracy of generated models and adaptability of the methodology. In the following, the proposed research questions are revisited and potential limitations of the methodology are discussed.

Research question 1 is answered by the results of analyzing available sources for dynamic production data in semiconductor manufacturing. It is shown that data sources that log material flow events (here lot tracking data) and production resource states (here RTC data) are well suited to mine the information for automatically a material flow simulation model. The fact that data of material flow events and

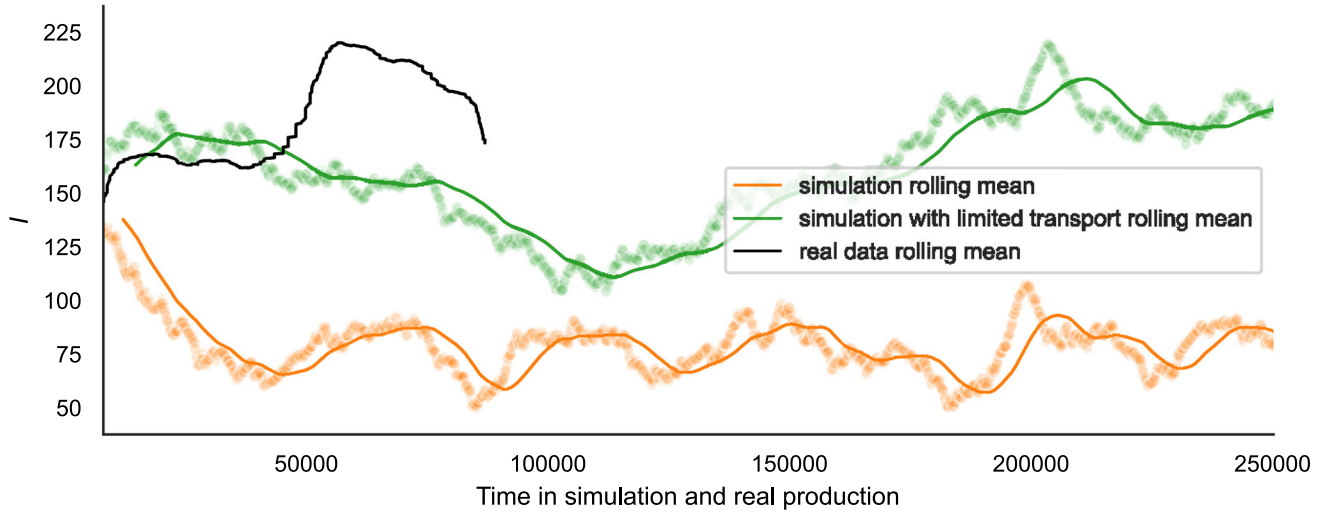

Fig. 18 Validation of inventory level distribution of the simulation model using a limited and unlimited number of transport resources with real production inventory level for the first use case

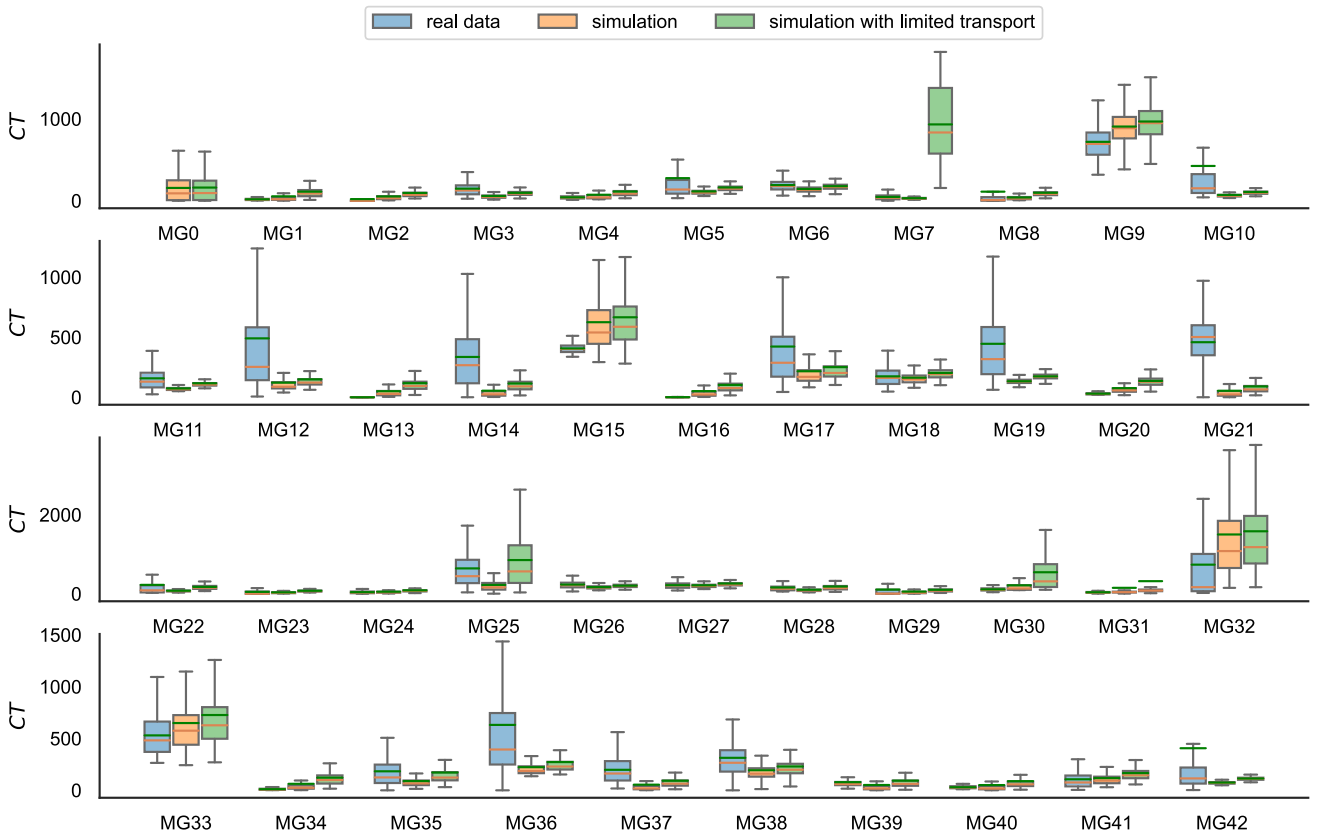

Fig. 19 Validation of cycle time distributions of machine groups of simulation model with real production cycle times of use case one using different limits (e.g. C13) on transport resources

production resource states is automatically gathered in most semiconductor manufacturing systems eliminates the dependency on manually maintained data and features a distinction from other approaches in literature that rely on manually maintained data. Since only two data sources are used with a very simple data structure, a transfer of the methodology to other environments should be possible with minimal adjustments of interfaces to these data sources.

The research demonstrates how to analyze this production data to generate simulation models of semiconductor production systems (research question 2). Based on implicitly provided information from two even data sources, information of the simulation model is extracted with process modeling, resource modeling and determination of cross influence. With this, the required time and effort for generation of simulation models of large production systems is dramatically reduced. Moreover, it is demonstrated how machine learning can be used as a solution for missing data. This indicates, that an integration of data-driven approaches, such as machine learning and data mining, in model-driven approaches, such as simulation, can be beneficial in the field of DT as adaptability can be increased and need for expert knowledge or manual efforts can be reduced. Moreover, the results show that relying only on statistical methods for data analysis may not be sufficient for ASMG due to low data accuracy or missing data.

Applying the methodology to two use cases—differing in modeled routes, the number of process steps, and tool compositions—demonstrates in regard of research question 3 that the methodology is universally applicable to automatically model semiconductor manufacturing production systems. Most assumptions that have been made for the presented ASMG methodology, such as the resource modeling classification scheme or the cross influence assumption, are generally valid in semiconductor manufacturing. Because similar data sources are common across many semiconductor manufacturers, only minor adaptations are typically needed for other environments.

The application of the methodology to use cases with production data of real production systems, however, shows that an extensive amount of data preparation is required due to data quality problems and outliers. Since data preparation is automated here, there remains a risk that important data points are removed during cleaning and filtering. For example, the outlier filtering removes data points that resemble abnormal behavior. This is necessary to generate valid time distributions but leads to a reduced variance in the modelled distributions in case of many outliers in the data. As the variance of a distribution influences the dynamic behavior of a production system, strongly filtered distributions can lead to wrong representations of the real production in the simulation model. However, filtering outliers allows to perform a target-performance comparison of the real production, as the simulation model resembles the real production but without extreme events and inefficiencies. This gives not only insights about the capability of the real production system but allows also for an evaluation of causes for the performance losses.

Potential limitations of the methodology include statistical assumptions made for simplifications (e.g. not considering auto-correlations of timely values) or due to missing data (e.g. considering auxiliary resources required for tool processing). These constraints can be overcome (1) by removing made simplifications if necessary and (2) incorporating additional data sources for more precise analyses. Trends like sensor fusion, enterprise data integration, and large language models (LLMs) in data integration could further advance ASMG capabilities.

With respect to simulation validity, the results from the two use cases confirm that the generated models accurately represent real production in terms of throughput, uptime, and utilization. Through equipment emulation and cross influence, a realistic load and utilization were achieved for most tools. However, without modeling a limited number of transport resources, the simulated cycle time is slightly lower than that of real production-likely due to missing transport time data and the removal of outliers. By constraining transport resources in the simulation, the cycle time approaches realistic levels. Future research should explore how outliers influence production systems and develop modeling procedures that incorporate them appropriately.

Some resources are modeled as "black boxes" using machine learning, meaning their internal logic may be less transparent than a first-principle approach. Although machine learning models capture dependencies in processing times, the reduced explainability could limit acceptance by machine experts. Techniques from explainable AI might help address this issue by providing clarity on model behavior. Concept drift can be mitigated by frequent retraining of machine learning models in the data mining pipeline. Future work could explore multi-model co-simulation, integrating physics-based simulation models for crucial components alongside machine learning approaches, thereby tackling confidence issues and ensuring higher fidelity in both modeling and simulation outputs.

Conclusion

This research contributes to the field of digital twins by closing the gap between real production systems and associated simulation models with a methodology that automatically generates a simulation model of a semiconductor manufacturing system by analyzing real production data with data mining and machine learning techniques. The results indicate, that the methodology is able to automatically extract all necessary information from production event logs for two

real use cases. Moreover, machine learning techniques are successfully utilized to classify and model complex cluster tools based on their historic processing behavior to reduce the need for explicit tool data. The assessment of the validity of the obtained simulation models by comparison to the real production shows that throughput, uptime and utilization are validly modelled. However, the observations indicate that some improvements considering inventory level and cycle time can be achieved by modeling the intra-logistics in the simulation more accurately. Moreover, further research could concentrate on a more generally applicable form to preprocess outliers and noise in production data and how the validity gap of simulation and real production could be overcome by optimizers for parameter tuning of the simulation.

Author Contributions Conceptualization: Andreas Kuhnle, Marvin May, Thomas Altenmüller and Gisela Lanza. Methodology: Sebastian Behrendt, Marvin May, Andreas Kuhnle, Thomas Altenmüller. Formal analysis and ivestigation: Sebastian Behrendt, Thomas Altenmüller. Writing—original draft preparation: Sebastian Behrendt. Writing—review and editing: Sebastian Behrendt, Andreas Kuhnle, Marvin May, Thomas Altenmüller and Gisela Lanza. Resources: Thomas Altenmüller and Andreas Kuhnle. Supervision: Thomas Altenmüller, Andreas Kuhnle and Gisela Lanza.

Funding Open Access funding enabled and organized by Projekt DEAL. No funding was received to assist with the preparation of this manuscript.

Data availability Due to confidentiality considerations, detailed information regarding the production data sets cannot be publicly disclosed.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

Bagchi, S., Chen-Ritzo, C.-H., Shikalgar, S. T., & Toner, M. (2008). A full-factory simulator as a daily decision-support tool for 300 mm wafer fabrication productivity. In S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, & J. W. Fowler (Eds.), *2008 Winter simulation conference* (pp. 2021–2029). IEEE.

Bengtsson, J., & Olhager, J. (2002). The impact of the product mix on the value of flexibility. *Omega, 30*(4), 265–273. https://doi.org/10.1016/S0305-0483(02)00034-8

Bergmann, S., & Strassburger, S. (2010). Challenges for the automatic generation of simulation models for production systems. In *Proceedings of the 2010 summer computer simulation conference* (pp. 545–549).

Camargo, M., Dumas, M., & Rojas, O. G. (2020). Discovering generative models from event logs: Data-driven simulation vs deep learning. *CoRR*. https://doi.org/10.48550/arXiv.2009.03567

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In B. Krishnapuram, M. Shah, A. Smola, C. Aggarwal, D. Shen, & R. Rastogi (Eds.), *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 785–794). ACM.

Choudhary, A. K., Harding, J. A., & Tiwari, M. K. (2009). Data mining in manufacturing: A review based on the kind of knowledge. *Journal of Intelligent Manufacturing, 20*(5), 501–521. https://doi.org/10.1007/s10845-008-0145-x

Denno, P., Dickerson, C., & Harding, J. A. (2018). Dynamic production system identification for smart manufacturing systems. *Journal of Manufacturing Systems, 48*, 192–203. https://doi.org/10.1016/j.jmsy.2018.04.006

Dudin, A. N., Klimenok, V. I., & Vishnevsky, V. M. (2020). *The theory of queuing systems with correlated flows*. Springer.

Eckardt, F. (2002). *Ein beitrag zu theorie und praxis datengetriebener modellgeneratoren zur simulation von produktionssystemen*. Shaker.

Farsi, M., Daneshkhah, A., Hosseinian-Far, A., & Jahankhani, H. (2020). *Digital twin technologies and smart cities* (1st ed.). Springer.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics, 29*(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

Göppert, A., Grahn, L., Rachner, J., Grunert, D., Hort, S., & Schmitt, R. H. (2023). Pipeline for ontology-based modeling and automated deployment of digital twins for planning and control of manufacturing systems. *Journal of Intelligent Manufacturing, 34*(5), 2133–2152. https://doi.org/10.1007/s10845-021-01860-6

Grieves, M. (2014). *Digital twin: Manufacturing excellence through virtual factory replication* (Whitepaper). Florida Institute of Technology. https://www.3ds.com/fileadmin/PRODUCTS-SERVICES/DELMIA/PDF/Whitepaper/DELMIA-APRISO-Digital-Twin-Whitepaper.pdf

Keller, A., Kamath, A., & Perera, U. (1982). Reliability analysis of CNC machine tools. *Reliability Engineering, 3*(6), 449–473. https://doi.org/10.1016/0143-8174(82)90036-1

Kohn, R., & Werner, S. (2010). Automated semiconductor equipment modeling and model parameter estimation using MES data. In J. Barnum & D. Maynard (Eds.), *2010 IEEE/semi advanced semiconductor manufacturing conference (ASMC)* (pp. 11–16). IEEE.

Kritzinger, W., Karner, M., Traar, G., Henjes, J., & Sihn, W. (2018). Digital twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine, 51*(11), 1016–1022. https://doi.org/10.1016/j.ifacol.2018.08.474

Kuhnle, A. (2020). *Adaptive order dispatching based on reinforcement learning: Application in a complex job shop in the semiconductor industry* (1st ed., Vol. 241). Shaker.

Kusiak, A. (2023). Manufacturing metaverse. *Journal of Intelligent Manufacturing, 34*(6), 2511–2512. https://doi.org/10.1007/s10845-023-02145-w

Lasi, H., Fettke, P., Kemper, H.-G., Feld, T., & Hoffmann, M. (2014). Industry 4.0. *Business & Information Systems Engineering, 6*(4), 239–242. https://doi.org/10.1007/s12599-014-0334-4

Lee, J., Azamfar, M., Singh, J., & Siahpour, S. (2020). Integration of digital twin and deep learning in cyber-physical systems: Towards

smart manufacturing. *IET Collaborative Intelligent Manufacturing, 2*(1), 34–36. https://doi.org/10.1049/iet-cim.2020.0009

Lee, J. Y., Kang, H. S., Kim, G. Y., & Noh, S. D. (2012). Concurrent material flow analysis by P3R-driven modeling and simulation in PLM. *Computers in Industry, 63*(5), 513–527. https://doi.org/10.1016/j.compind.2012.02.004

Lee, T.-E. (2008). A review of scheduling theory and methods for semiconductor manufacturing cluster tools. In S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, & J. W. Fowler (Eds.), *2008 Winter simulation conference* (pp. 2127–2135). IEEE.

Lingitz, L., Gallina, V., Ansari, F., Gyulai, D., Pfeiffer, A., Sihn, W., & Monostori, L. (2018). Lead time prediction using machine learning algorithms: A case study by a semiconductor manufacturer. *Procedia CIRP, 72*, 1051–1056. https://doi.org/10.1016/j.procir.2018.03.148

Little, J. D. C. (1961). A proof for the queuing formula: $L = \lambda W$. *Operations Research, 9*(3), 383–387. http://www.jstor.org/stable/167570

Martínez, G. S., Sierla, S., Karhela, T., & Vyatkin, V. (2018). Automatic generation of a simulation-based digital twin of an industrial process plant. In *IECON 2018—44th annual conference of the IEEE industrial electronics society* (pp. 3084–3089).

May, L. O. M. C., Nestroy, Christian, & Lanza, G. (2023). Automated model generation framework for material flow simulations of production systems. *International Journal of Production Research, 62*(1–2), 141–156. https://doi.org/10.1080/00207543.2023.2284833

May, M. C., Overbeck, L., Wurster, M., Kuhnle, A., & Lanza, G. (2021). Foresighted digital twin for situational agent selection in production control. *Procedia CIRP, 99*, 27–32. https://doi.org/10.1016/j.procir.2021.03.005

Milde, M., & Reinhart, G. (2019). Automated model development and parametrization of material flow simulations. In *2019 Winter simulation conference (WSC)* (pp. 2166–2177). IEEE Press.

Mönch, L., Fowler, J. W., Dauzère-Pérès, S., Mason, S. J., & Rose, O. (2011). A survey of problems, solution techniques, and future challenges in scheduling semiconductor manufacturing operations. *Journal of Scheduling, 14*(6), 583–599. https://doi.org/10.1007/s10951-010-0222-9

Mönch, L., Fowler, J. W., & Mason, S. J. (2012). *Production planning and control for semiconductor wafer fabrication facilities: Modeling, analysis, and systems* (Vol. 52). New York: Springer.

Mourtzis, D., Doukas, M., & Bernidaki, D. (2014). Simulation in manufacturing: Review and challenges. *Procedia CIRP, 25*, 213–229. https://doi.org/10.1016/j.procir.2014.10.032

Negri, E., Fumagalli, L., & Macchi, M. (2017). A review of the roles of digital twin in CPS-based production systems. *Procedia Manufacturing, 11*, 939–948. https://doi.org/10.1016/j.promfg.2017.07.198

Oliveira, M. S. D., Santos, C. H. D., Gabriel, G. T., Leal, F., & Montevechi, J. A. B. (2023). Famosim: A facilitated discrete event simulation framework to support online studies. *Production, 33*, e20220073. https://doi.org/10.1590/0103-6513.20220073

Overbeck, L., Le Louarn, A., Brützel, O., Stricker, N., & Lanza, G. (2021). Continuous validation and updating for high accuracy of digital twins of production systems. In *Simulation in produktion und logistik 2021* (Vol. 0, pp. 609–617). Cuvillier Verlag.

Rasheed, A., San, O., & Kvamsdal, T. (2020). Digital twin: Values, challenges and enablers from a modeling perspective. *IEEE Access, 8*, 21980–22012. https://doi.org/10.1109/ACCESS.2020.2970143

Rathore, M. M., Shah, S. A., Shukla, D., Bentafat, E., & Bakiras, S. (2021). The role of AI, machine learning, and big data in digital twinning: A systematic literature review, challenges, and opportunities. *IEEE Access, 9*, 32030–32052. https://doi.org/10.1109/ACCESS.2021.3060863

Reinhardt, H., Weber, M., & Putz, M. (2019). A survey on automatic model generation for material flow simulation in discrete manufacturing. *Procedia CIRP, 81*, 121–126. https://doi.org/10.1016/j.procir.2019.03.022

Shanthikumar, J. G., Ding, S., & Zhang, M. T. (2007). Queueing theory for semiconductor manufacturing systems: A survey and open problems. *IEEE Transactions on Automation Science and Engineering, 4*(4), 513–522. https://doi.org/10.1109/TASE.2007.906348

Tliba, K., Diallo, T. M. L., Penas, O., Ben Khalifa, R., Ben Yahia, N., & Choley, J.-Y. (2023). Digital twin-driven dynamic scheduling of a hybrid flow shop. *Journal of Intelligent Manufacturing, 34*(5), 2281–2306. https://doi.org/10.1007/s10845-022-01922-3

Tukey, J. W. (1977). *Exploratory data analysis*. Addison-Wesley.

van der Aalst, W. (2011). *Process mining: Discovery, conformance and enhancement of business processes*. Springer.

Vernickel, K., Brunner, L., Hoellthaler, G., Sansivieri, G., Härdtlein, C., Trauner, L., & Berg, J. (2020). Machine-learning-based approach for parameterizing material flow simulation models. *Procedia CIRP, 93*, 407–412. https://doi.org/10.1016/j.procir.2020.04.018

von Rueden, L., Mayer, S., Sifa, R., Bauckhage, C., & Garcke, J. (2020). Combining machine learning and simulation to a hybrid modelling approach: Current and future directions. In M. R. Berthold, A. Feelders, & G. Krempl (Eds.), *Advances in intelligent data analysis xviii. IDA 2020* (Vol. 12080, pp. 548–560). Springer.

Waschneck, B., Altenmüller, T., Bauernhansl, T., & Kyek, A. (2016). Production scheduling in complex job shops from an industry 4.0 perspective: A review and challenges in the semiconductor industry. In *Sami40 workshop at I-know 2016.* (pp. 1–12). Graz, 18.10-19.10.2016.

White, K. P. (1997). An effective truncation heuristic for bias reduction in simulation output. *SIMULATION, 69*(6), 323–334. https://doi.org/10.1177/003754979706900601

Wuest, T., Weimer, D., Irgens, C., & Thoben, K.-D. (2016). Machine learning in manufacturing: Advantages, challenges, and applications. *Production & Manufacturing Research, 4*(1), 23–45. https://doi.org/10.1080/21693277.2016.1192517