Proceedings of the 2024 Winter Simulation Conference
H. Lam, E. Azar, D. Batur, S. Gao, W. Xie, S. R. Hunter, and M. D. Rossetti, eds.

SYNTHETIC SIMULATED ENVIRONMENT FOR DISCRETE MANUFACTURING SYSTEMS: A DEMONSTRATOR THROUGH A COMPUTATIONAL MODELING APPROACH

Silvan Marti[1], Paulo Victor Lopes[1,2], Siyuan Chen[1], Mohan Rajashekarappa[1], Elham Rekabi Bana[1], Amon Göppert[3], Mélanie Déspeisse[1], Johan Stahre[1], and Björn Johansson[1]

[1]Dept. of Industrial and Materials Science, Chalmers University of Technology, Gothenburg, SWEDEN
[2]Dept. Computer Science, Aeronautics Institute of Technology, São José dos Campos, BRAZIL
[3]Lab. for Machine Tools and Production Eng. (WZL), RWTH Aachen University, Aachen, GERMANY

ABSTRACT

In light of the challenges posed by the often unavailability of coherent data in manufacturing for operational Artificial Intelligence (AI) decision support systems, the generation and utilization of synthetic datasets have become essential. This study introduces a simple numerical Synthetic Simulated Environment (SSE) using timed and parametrizable Petri Net (PN) modules, embedded in a Directed Acyclic Graph (DAG) structure described by an adjacency matrix to represent material flow. Implemented in PyTorch for seamless integration with AI components, our simulation framework simplifies manufacturing systems, yet remains expandable for diverse use cases. The simulation model was demonstrated displaying its capability of generating synthetic data. This approach explores the practicality and applicability of generated data. It could serve as an ideal environment to benchmark Artificial Intelligence (AI) algorithms in comparative experiments, investigating operational problems featured in the dynamic interactions of discrete manufacturing systems.

1 INTRODUCTION

Discrete manufacturing systems are focused on the production of distinct items that can be counted, such as automobiles, electronics, and furniture. Each product in discrete manufacturing is made through a series of individual steps. There are some operational challenges which hinder efficient manufacturing operations, such as bottlenecks (Rocha and Lopes 2022) or equipment downtime (Saez et al. 2018). However, these disruptions can be mitigated and operational efficiency improved by supervising and analyzing a process in real-time (Saez et al. 2018).

Supervising a system is possible by data collection through Internet of Things (IoT) devices which can collect vast amounts of data. They are broadly applied to monitor vitals of diverse systems in healthcare, smart homes, and also manufacturing (Anderson et al. 2014). Much of this data is represented as time series measurements which can be analyzed using parametric artificial intelligence methods, e.g., Deep Learning (DL) for downstream tasks such as anomaly detection (Liu et al. 2023), forecasting (Yin et al. 2019), and causal inference (Gong et al. 2023). Real-time anomaly detection in discrete manufacturing can, for instance, lead to better fault detection or condition monitoring (Jeong et al. 2022).

However, analyzing time series data involves overcoming several challenges. The collected data can be very high-dimensional. For instance, a modern refinery may feature up to a million sensors each tracking various parameters in real-time (Wang and Gross 2018). For deep learning time series models, this characteristic poses a challenge which can be met with adequate scalability of the models towards the number of time series. Another challenge in IoT data is that time series can be of non-stationary distributions (Liu et al. 2023). Lastly, in discrete manufacturing, data associated with product flow is featured not as continuous valued time series but also as discrete valued series such as stock and buffer levels. This is in combination with other discrete data such as machine operation states (Saez et al. 2018).

This type of data series, which is composed of non-negative integer values over time, has great importance in a variety of fields including finance and earthquake-occurrence modeling (Manolakis et al. 2019). Benchmark datasets play a crucial role in evaluating algorithmic performance (Thiyagalingam et al. 2022). To investigate data analysis in these count series, a number of datasets are openly accessible. For instance MIMIC-III (Medical Information Mart for Intensive Care) provides extensive, time-stamped patient health records that include counts of medical events such as admissions, lab tests, and administered medications (Johnson et al. 2016). However, in specialized domains like discrete manufacturing production flows, comprehensive datasets that thoroughly analyze the complex interactions of machines are notably lacking, to the best of the authors' knowledge.

In various fields, Synthetic Data Generation (SDG) techniques have proven invaluable for simulating complex scenarios where acquiring real-world data is impractical or costly (Nikolenko 2021). SDG is broadly applied in computer vision for supervised training and testing machine learning algorithms (Paulin and Ivasic-Kos 2023). For instance, SDG is employed in self-driving car research to create simulated environments that replicate real-world conditions, including virtual roads, traffic dynamics, pedestrians, and dynamic obstacles (Kaur et al. 2021). These simulations allow researchers to test and refine algorithms under diverse scenarios to assess performance without real-world risks. Moreover, beyond applications in computer vision and autonomous systems, SDG techniques are gaining traction in biomedical research. Researchers utilize synthetic data to generate realistic patient datasets that facilitate the development and validation of predictive models and treatment strategies (Goncalves et al. 2020).

In this regard, to study the scalability of deep learning time series analysis models, we propose to generate an arbitrary number of non-independent discrete time series using a simulation approach. The system-level dynamics of discrete manufacturing are generally well understood and are often studied using a Discrete Event Systems (DES) approach (Saez et al. 2018). There are a multitude of simulation models available. Markov models are powerful for modeling systems where future states depend only on the current state, making them suitable for time series analysis with well-defined state transitions (Cassandras and Lafortune 2021). In contrast, Petri Nets offer a more comprehensive representation by explicitly modeling concurrent events and dependencies, making them better suited for simulating the dynamic behavior of manufacturing processes. To study the discrete manufacturing production flow dynamics, Petri Nets (PNs) are frequently used to model the concurrent dynamics. Petri Nets allow explicit modeling of concurrent events and resource interactions, providing a more detailed representation of complex workflows and decision-making processes within production systems (Cassandras and Lafortune 2021).

1.1 Objective

The main objective of this paper is to propose a scalable simulation of a discrete manufacturing system as a parameterizeable Synthetic Simulated Environment (SSE). The purpose of this simulation is to represent machine variables as non independent and nonegative integer time series. To achieve this main objective, the paper will aim for these specific objectives:

1. Propose a simplified model of a production unit which in numbers can be scaled to resemble the complex interactions of a production line.
2. Describe the mathematical foundation for the numerical simulation model.
3. Demonstrate the capability of the proposed model in a practical use case.

In Section 2, the paper describes some related work on SDG, time series and discrete manufacturing approaches. Section 3 proposes the methodology based on the concept of Petri Nets, Finite State Machines and probability of events. The results are presented and discussed in Section 4, showcasing the data generating capability in a practical example.

2 RELATED WORK

In manufacturing, synthetic data can be generated through empirical approaches and simulation techniques designed to replicate dynamic behaviors (Libes et al. 2017). Empirical methods involve constructing laboratory setups, such as a production line built using LEGO®, to study and generate model processes (Lugaresi and Matta 2021). However, due to scalability needs, our focus primarily centers on simulation-based approaches.

2.1 Simulation Approaches for Manufacturing

Discrete event simulations (DES) are widely employed in discrete manufacturing processes. A hybrid model combining DES and Continuous Dynamics (CD) can simulate individual machine behaviors and system-level interactions in real-time, ensuring synchronization between virtual and physical environments (Saez et al. 2018). Markov models, which generate time series data by transitioning between defined states with set probabilities, are another method used for simulation (Shamshad et al. 2005). However, both approaches may struggle with larger systems' scalability. Digital twin simulations, unlike DES and Markov models, can represent various dynamics, offering a comprehensive view of continuous and discrete interactions (Phanden et al. 2021). However, for concurrent discrete dynamics like product flow, Petri Nets provide a comprehensive framework.

2.2 Petri Nets in Manufacturing Systems

Petri Nets offer a detailed and scalable modeling approach for complex manufacturing systems (Cassandras and Lafortune 2021). Various extensions have been developed to address specific needs:

- **Colored Petri Nets (CPNs):** Extend traditional Petri Nets by incorporating data, enabling more detailed and scalable modeling through tokens that carry additional information such as attributes or values (Gehlot and Nigro 2010).
- **Hierarchical Petri Nets (HPNs):** Enhance traditional Petri Nets by decomposing complex systems into nested subnets, facilitating modular and scalable modeling (Fehling 1993).
- **Hierarchical Timed Colored Petri Nets (HTCPNs):** Model complex distributed manufacturing systems by representing product flow and transitions, addressing the need for improved throughput in modern manufacturing networks (Lv et al. 2013).
- **Timed Petri Nets:** Analyze system behavior within manufacturing systems, representing asynchronous and concurrent transactions (Reddy et al. 1993). They are used for operational decisions, such as scheduling (Casalino et al. 2021) or dispatching (Hu and Liu 2015).

Recent research has integrated Petri Nets with reinforcement learning (RL) algorithms to tackle job shop scheduling problems (Lassoued and Schwung 2024). Parametrizable flow simulation models have also shown promise in automating and streamlining labor-intensive tasks like model development, as demonstrated by Milde and Reinhart (Milde and Reinhart 2019).

2.3 Conclusion of Related Models

Existing manufacturing simulation implementations often represent a given manufacturing system to study a specific problem. These implementations typically focus on analyzing static or well-defined systems to address particular issues such as optimizing a production line. However, for generating scalable time series data of dynamic material flows, they are limited in suitability for synthetic data generation.

This study introduces a novel approach using Timed Petri Nets within a Directed Acyclic Graph (DAG) framework inspired by HPNs. The simulation model is implemented in PyTorch (Paszke et al. 2019) for simple AI integration. This method aims to streamline system modeling, minimize parameterization, and enhance simulation efficiency for comprehensive manufacturing studies.

3 METHODS

This study introduces a simplified numerical SSE employing timed and configurable PN modules, representing individual manufacturing assets within a Directed Acyclic Graph (DAG) structure characterized by an adjacency matrix for material flow depiction. The implementation, carried out in PyTorch (Paszke et al. 2019), is chosen for its seamless integration potential with Artificial Intelligence (AI) components for subsequent simulation-based studies. While our simulation framework streamlines manufacturing system modeling by representing it as a DAG without rework loops, its open-source nature allows for extensibility to cater to diverse use cases. Additionally, our research endeavors to construct a concise meta-model of a discrete manufacturing system, focusing on minimal parameters for efficient data generation. The model adopts a discrete simulation approach, featuring entities with unique attributes that influence simulation states, interactions with activities triggering events leading to state changes, resources with limited capacity, and global variables for universal accessibility and key metric tracking. Random number generators facilitate sampling, while a calendar schedules future events. Essential system state variables, such as current time, and statistics collectors, which monitor states or entity attributes, complete the model's key components.

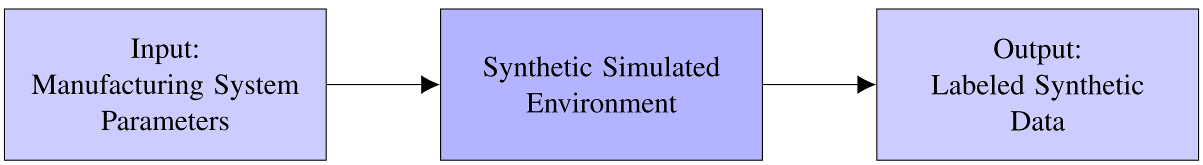


Figure 1: The workflow to generate labeled synthetic data.

As visible in Figure 1, the simulation model is defined by the input parameters such as the adjacency matrix of a manufacturing system, set up parameters such as buffer limits and distributions for timed events. In this section we delve into the general concept of the used techniques in Section 3.1 and present the resulting implementation in Section 3.2.

3.1 Simulation Concept

We have defined a simplified discrete manufacturing system as DAG. A graph is represented as a set of nodes $V, |V| = n$ and a set of edges $E, |E| = m$. The system can be denoted as $G = (V,E)$. The relations between node v_i and node v_j can be represented as an adjacency matrix $A_{ij} \in \mathbb{N}^{n\times n}$, where each entry A_{ij} indicates the number of direct edges from node v_i to node v_j. If there is no direct edge between v_i and v_j, then $A_{ij} = 0$. A graph is directed if $A \neq A^T$ and acyclic if there are no directed circuits (Chen 2012). newline

Graph based product flow: Each node v_i contains a PN module as illustrated in Figure 2. Additionally, when nodes v_i and v_j are connected, not only is the adjacency matrix A_{ij} updated to reflect this connection, but a weight ω_{ij} is also assigned to each connection. This weight represents the number of parts that need to be transported simultaneously between the nodes, thus adding a layer of quantitative detail to the model that informs the logistical or operational capacities required for material movement between processes.

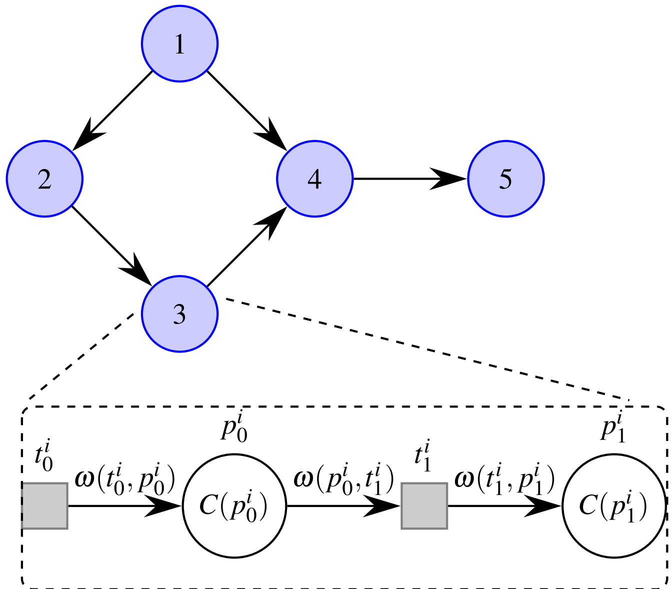


Figure 2: A combined figure illustrating a directed acyclic graph (DAG) modeling a discrete manufacturing system with repositioned nodes (above). Each node v_i in the network includes a Petri net module (below).

Our model simplifies the representation of the entire system to a degree equivalent to a traditional PN. However, a key advantage of our approach lies in the significantly reduced dimensionality of the resulting adjacency matrix of the DAG compared to the matrix of flow arcs in an equivalent PN. This reduction in complexity is particularly beneficial for simulation purposes and for generating labeled data for representation learning.

3.1.1 Petri Net Module

In our nested implementation displayed in Figure 2 we placed a PN inside each production node v_i. A PN is a mathematical modeling tool used to describe and analyze systems involving concurrent events and shared resources. It is defined as a tuple (P,T,F,ω), where $P, |P| = n_{pn}$ represents places for tokens, $T, |T| = m$ a set of transitions, $F \subseteq (P \times T) \cup (T \times P)$ flow arcs, and $\omega : F \rightarrow \{1,\ldots\}$ a weight function on these arcs.

Central to a PN's dynamics is the concept of markings, $M : P \rightarrow \mathbb{N}$, which assign a non-negative integer to each place indicating the token count. The system state is thus defined.

When a transition t fires, it alters markings based on the rule:

$$M'(p) = M(p) - \omega(p,t) + \omega(t, p') \text{ for all connected } p, p' \in P, \tag{1}$$

where M' represents the new state post-transition. This marking update is critical for analyzing the Petri Net's behavior (Cassandras and Lafortune 2021).

Timed Petri Net Module: We have defined the PN-module illustrated in Figure 2 as follows:

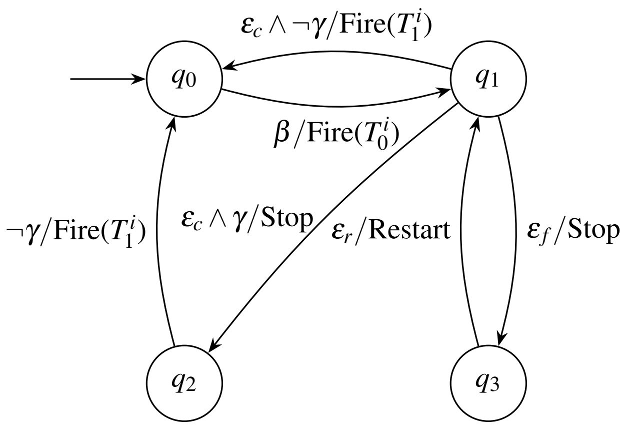
$$\begin{aligned} PN_{\text{module } v_i} = (P &= \{p_0, p_1\}, \\ T &= \{t_0, t_1\}, \\ F &= \{(t_0, p_0), (p_0, t_1), (t_1, p_1)\}, \\ \omega &= \{\omega(t_0, p_0), \omega(p_0, t_1), \omega(t_1, p_1) = \omega(p_0, t_1)\}) \end{aligned} \tag{2}$$

t_0 and t_1 transitions manage token flows between production (p_0) and output buffer (p_1) spaces, respectively, which are constrained by capacities $C(p_0)$ and $C(p_1)$, reflecting the operational limits and interactions of the network. This representation highlights the intricate, interconnected nature of manufacturing processes.

3.1.2 Manufacturing Parameters and Finite State Machine

Determining the properties of a simulation model is crucial for accurately predicting manufacturing outcomes and enhancing system design and optimization. We compile essential parameters that describe industrial assets, categorizing them into temporal parameters, quality metrics, and performance indicators. These parameters play a vital role in the evaluation and improvement of manufacturing processes.

Building on our objectives, we implement an Finite State Machine (FSM) for each node v_i to simulate the operational workflow of a production system asset. A FSM is a computational model used to design both computer programs and sequential logic circuits, consisting of a finite number of states. It is defined by a tuple $(S, S_0, \Sigma, \delta, F)$, where S is a finite set of states, $S_0 \subseteq S$ includes the initial states, Σ is a set of input symbols (input alphabet), $\delta : S \times \Sigma \to S$ is the state transition function, and $F \subseteq S$ represents the set of final or accepting states.



(a)

Category	Symbol	Description
States	q_0	Idle
	q_1	Producing
	q_2	Blocked
	q_3	Failed
Events	ε_c	Cycle Time Reached
	ε_f	Time to Failure Reached
	ε_r	Time to Repair Reached
	β	Input part available
	γ	Output space available

(b)

Figure 3: Illustration of the Mealy machine (a) used to model the operational states of a production system asset. Each transition is labeled with an event and its corresponding output, reflecting the immediate effect of the transition. Each symbol is explained in (b).

Our model defines the asset's potential states with $S = \{q_0, q_1, q_2, q_3\}$, each representing a distinct operational condition, which are detailed in Figure 3 (a), including the state transition functions. The set of input symbols, $\Sigma = \{\varepsilon_c, \varepsilon_f, \varepsilon_r, \beta, \gamma\}$, includes triggers for transitions between these states, as specified in Figure 3 (b). The initial state of our system is set as $s_0 = q_0$.

3.1.3 Probability Distributions of Temporal Events

In an event-driven model, transitions between states are not governed by fixed time intervals but by the occurrences of discrete events. We denote the current system time by τ_{current}, and define $\mathscr{E} = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n\}$ as the set of all planned events. Each event ε_i in this set is scheduled at a time τ_i where $\tau_i \geq \tau_{\text{current}}$. The system transitions to the state associated with the next event $\varepsilon_{\text{next}}$, determined by (Cassandras and Lafortune 2021):

$$\tau_{\text{next}} = \min\{\tau_i | \tau_i > \tau_{\text{current}}, \varepsilon_i \in \mathscr{E}\} \in \mathbb{R}^+ \tag{3}$$

Temporal Event Randomization: Expanding upon this model, each node v_i in our system experiences temporal events $\varepsilon_p^i, \varepsilon_f^i,$ and ε_r^i which are strictly dependent on their respective occurrence times $\tau_p, \tau_f,$ and τ_r aligning with τ_{current}. These event occurrences are modeled as random variables:

$$\varepsilon_x^i \sim \mathscr{F}_x(\tau_x) \quad \forall x \in \{p, f, r\}, \forall i \in V, \tag{4}$$

where $\mathscr{F}_x$ is an unknown distribution function independently assigned to each type of event at each node. This setup ensures that the event timings τ_x are independent and identically distributed across different nodes and event types, introducing a level of stochastic variability and unpredictability inherent in the system's dynamics. These independent distributions may exhibit a variety of characteristics.

3.2 System Dynamics

The dynamic behaviors of discrete manufacturing systems are fundamentally influenced by their underlying variables and operational rules. This section outlines the key variables of our system model.

3.2.1 System Variables Overview

The following table provides a concise overview of the key matrices utilized in the system model, detailing their structure and purpose:

Table 1: An overview of system matrices which are used in the implementation.

Matrix	Dimensions	Data Type	Definition
State Matrix (Q)	$n \times 4$	$\mathbb{N}$	$q_{ij} = \begin{cases} 1 & \text{if } v_i \text{ is in state } q_j, \\ 0 & \text{otherwise} \end{cases}$
Marking Matrix (M)	$n \times 2$	$\mathbb{N}$	$m_{ij} = M(p_j^i)$
Event Timing Matrix (E)	$n \times 3$	$\mathbb{R}^+$	$e_{ij} = \begin{cases} \tau_p^i & \text{if } j = 1, \\ \tau_f^i & \text{if } j = 2, \\ \tau_r^i & \text{if } j = 3 \end{cases}$
Capacity Matrix (C)	$n \times 2$	$\mathbb{N}$	$c_{ij} = \begin{cases} C(p_0^i) & \text{if } j = 1, \\ C(p_1^i) & \text{if } j = 2 \end{cases}$
Weight Matrix (W)	$n \times 2$	$\mathbb{N}$	$w_{ij} = \begin{cases} \omega(t_0^i, p_0^i) & \text{if } j = 1, \\ \omega(t_1^i, p_1^i) & \text{if } j = 2 \end{cases}$
Adjacency Matrix (A)	$n \times n$	$\mathbb{N}$	$A_{ij} = \begin{cases} \omega_{ij} & \text{if } v_i \text{ and } v_j \text{ directly con. }, \\ 0 & \text{otherwise} \end{cases}$

3.2.2 Algorithm

This algorithm is designed to optimize the execution of events by selecting the most imminent event based on the current state of the system. It handles the complexity of event scheduling by determining which event should occur next and precisely when it should be executed to maintain the integrity and accuracy of the system's operational timeline.

Algorithm 1 Event Scheduling and Simulation Algorithm

Find Minimum Event Time:
$E_{\min} \leftarrow \min(\{E_{ij} \mid Q_i\})$ ▷ Find the smallest event time regarding the states
$(i^*, j^*) \leftarrow \arg\min_{i,j}(E_{ij})$ ▷ Identify the node and event type with the minimum time
Define Event Variable:
$\text{event} \leftarrow \begin{cases} \varepsilon_p & \text{if } j^* = 0 \\ \varepsilon_f & \text{if } j^* = 1 \\ \varepsilon_r & \text{if } j^* = 2 \end{cases}$ ▷ Determine the event based on the index
Execute Event:
Execute event at node i^*
Update state $\mathbf{Q}_i$
Transition Nodes which are in state 0 and 2
while any(node k^* unblocked or supplied) **do**
    Update state $\mathbf{Q}_k$
    Fire Transitions t_k and update $\mathbf{M}$
end while

4 DEMONSTRATION OF TIME SERIES DATA

In this demonstration, we visualize data from a simulated manufacturing system with 5 nodes illustrated in Figure 4. Using color-coded plots, we illustrate the buffer levels and operational states over time. The mean values of τ_c, τ_f, and τ_r in the manufacturing system are 180s, 18000s, and 3600s, respectively. These parameters, denoted as temporal variables governing the system, follow normal distributions $\mathcal{N}(180, 72)$, $\mathcal{N}(18000, 7200)$, and $\mathcal{N}(3600, 1440)$, respectively. Additionally, the capacities $C(p_0)_i$ and $C(p_1)_i$ for each node are set to 1 and 5, respectively, except for the last node, which has an unlimited capacity as the final output.

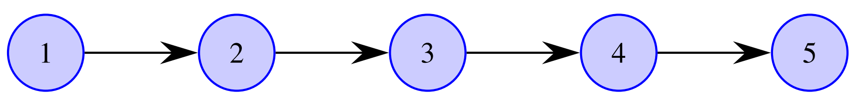


Figure 4: A simple manufacturing system which is used as a demonstrator.

In Figure 5, the buffer levels and final produced number of products are displayed, showcasing the simulation data generation process. The buffer levels represent the amount of inventory at each node throughout the simulation, while the final produced number of products illustrates the overall output of the manufacturing system.

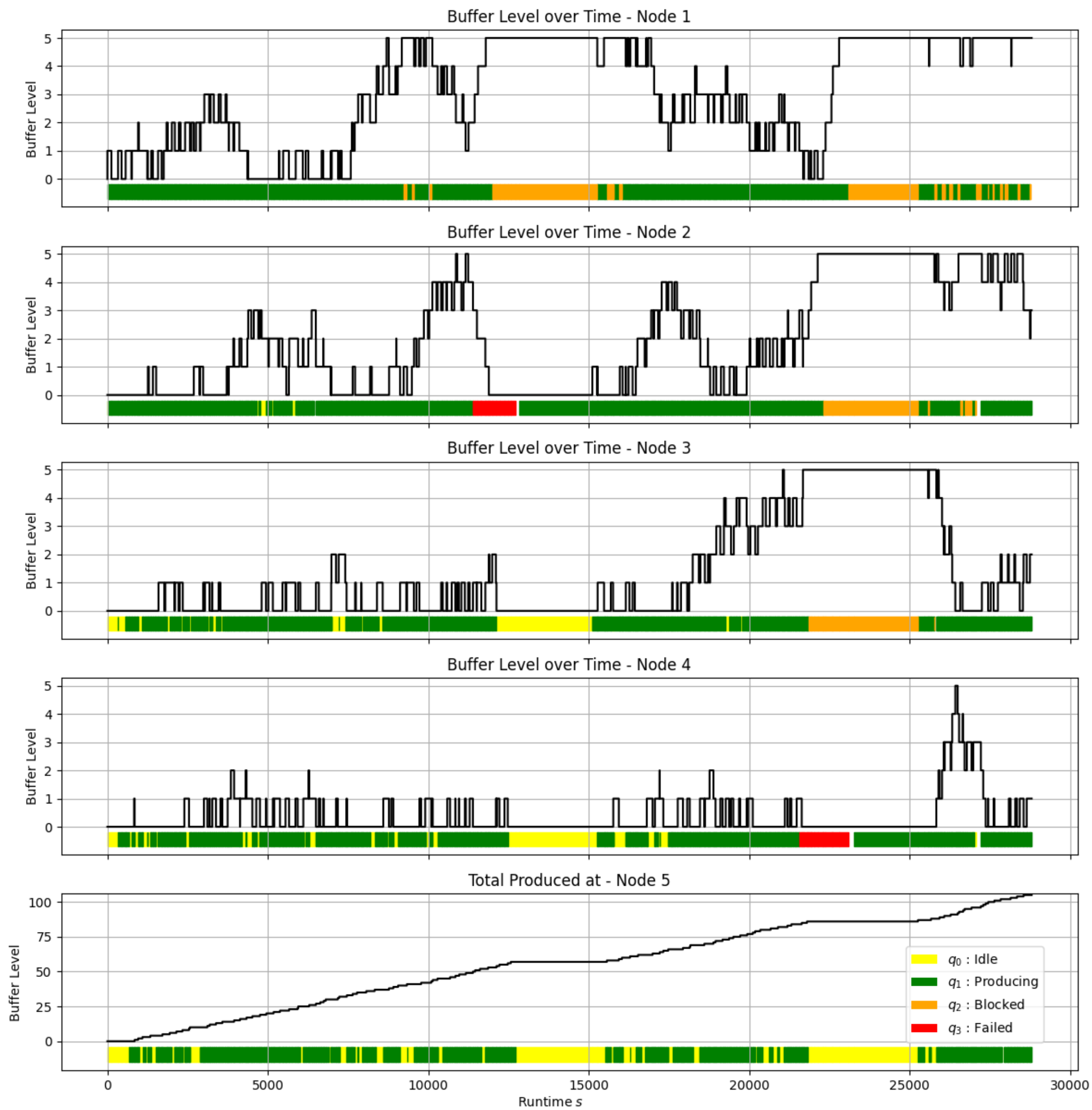


Figure 5: The visualization of buffer levels and machine states as they evolve over time in a manufacturing system.

4.1 Discussion

Our approach focuses on generating time series data as vectorized representations, which are commonly used in deep learning networks for efficient processing and analysis. This contrasts with the method proposed for instance by Anderson (Anderson et al. 2014), who generates synthetic data by creating XML files representing log files. While Anderson's approach aims at providing a comprehensive log file structure for various applications, our method specifically produces time series data suitable for deep learning models, ensuring a streamlined and focused dataset for AI-based analysis.

The experiment demonstrates the simulated manufacturing system's capabilities. Our framework can simulate a system of n nodes (machines) with given buffer limits, creating an n-dimensional dataset of buffer levels over time. Additionally, it generates an n-dimensional dataset of machine states as time series. However, further experiments with real data are necessary to enhance the models validity for future AI model training.

The model is currently limited as it does not including cyclic process flows and rework loops, however they are essential for industries like semiconductors. Expanding the model to incorporate rework loops will require modifying the Petri Net structure. Further, maintaining the model's parallel computing capabilities is crucial for fast computation and scalability. This aspect is crucial for enabling efficient simulation of large-scale manufacturing systems and supporting real-time, high-volume data generation.

5 CONCLUSION AND FURTHER WORK

In conclusion, our study highlights the importance of synthetic datasets for overcoming data scarcity in manufacturing AI systems. We introduced a Synthetic Simulated Environment (SSE) using timed and parametrizable Petri Net (PN) modules within a Directed Acyclic Graph (DAG) framework. Implemented in PyTorch, this framework offers a flexible and efficient way to represent material flow dynamics and generate data for diverse applications.

While traditional simulation models focus on static systems, our approach generates scalable time series data of dynamic material flows, better suited for synthetic data generation. We plan to modify the Petri Net structure to support cycles and rework. This could further improve the model to mimic distinct system dynamics of rework loops which are frequent in industries like semiconductors.

Additionally, we aim to extend our model to include both discrete and continuous time series, integrating variables like temperature and pressure with discrete events. Exploring domain randomization techniques, such as Generative Adversarial Networks (GANs), will further enhance dataset realism and variability, improving AI benchmarking in manufacturing systems.

By expanding these capabilities, we aim to provide a robust simulation framework that supports advanced time series analytics development, ultimately leading to enhanced manufacturing operations.

CODE AVAILABILITY

The code used for the analyses in this paper is available on GitHub at the following URL:

https://github.com/nachtflug6/ProcessTimeSeries

ACKNOWLEDGMENTS

This study was supported by European Union's Horizon 2020 research and innovation programme under grant no. 101058384 (RE4DY) and by the Swedish innovation agency VINNOVA under grant no. 2021-02421 (Digitala Stambanan). The work was carried out within Chalmers' Area of Advance Production whose support is greatly acknowledged.

REFERENCES

Anderson, J. W., K. E. Kennedy, L. B. Ngo, A. Luckow and A. W. Apon. 2014. "Synthetic Data Generation for the Internet of Things". In *2014 IEEE International Conference on Big Data (Big Data)*, 171–176: IEEE.

Casalino, A., A. M. Zanchettin, L. Piroddi, and P. Rocco. 2021. "Optimal Scheduling of Human–Robot Collaborative Assembly Operations with Time Petri Nets". *IEEE Transactions on Automation Science and Engineering* 18(1):70–84.

Cassandras, C. G. and S. Lafortune. 2021. *Introduction to Discrete Event Systems*. Cham: Springer International Publishing.

Chen, W.-K. 2012. *Applied Graph Theory*, Volume 13. Elsevier.

Fehling, R. 1993. "A Concept of Hierarchical Petri Nets with Building Blocks". In *Advances in Petri Nets 1993*, edited by G. Rozenberg, 148–168. Berlin, Heidelberg: Springer Berlin Heidelberg.

Gehlot, V. and C. Nigro. 2010. "An Introduction to Systems Modeling and Simulation with Colored Petri Nets". In *Proceedings of the 2010 Winter Simulation Conference*, 104–118 https://doi.org/10.1109/WSC.2010.5679170.

Goncalves, A., P. Ray, B. Soper, J. Stevens, L. Coyle and A. P. Sales. 2020. "Generation and Evaluation of Synthetic Patient Data". *BMC Medical Research Methodology* 20(1):108.

Gong, C., D. Yao, C. Zhang, W. Li and J. Bi. 2023. "Causal Discovery from Temporal Data: An Overview and New Perspectives".

Hu, H. and Y. Liu. 2015. "Supervisor Synthesis and Performance Improvement for Automated Manufacturing Systems by Using Petri Nets". *IEEE Transactions on Industrial Informatics* 11(2):450–458.

Jeong, K.-J., J.-D. Park, K. Hwang, S.-L. Kim and W.-Y. Shin. 2022. "Two-Stage Deep Anomaly Detection with Heterogeneous Time Series Data". *IEEE Access* 10:13704–13714.

Johnson, A. E. W., T. J. Pollard, L. Shen, L. w. H. Lehman, M. Feng, M. Ghassemi *et al*. 2016. "MIMIC-III, a Freely Accessible Critical Care Database". *Scientific Data* 3(1):1–9.

Kaur, P., S. Taghavi, Z. Tian, and W. Shi. 2021. "A Survey on Simulators for Testing Self-Driving Cars". In *2021 Fourth International Conference on Connected and Autonomous Driving (MetroCAD)*, 62–70: IEEE.

Lassoued, S. and A. Schwung. 2024. "Introducing PetriRL: An Innovative Framework for JSSP Resolution Integrating Petri Nets and Event-Based Reinforcement Learning". *Journal of Manufacturing Systems* 74:690–702.

Libes, D., D. Lechevalier, and S. Jain. 2017. "Issues in Synthetic Data Generation for Advanced Manufacturing". In *2017 IEEE International Conference on Big Data (Big Data)*, 1746–1754: IEEE.

Liu, Y., Y. Zhou, K. Yang, and X. Wang. 2023. "Unsupervised Deep Learning for IoT Time Series". *IEEE Internet of Things Journal* 10(16):14285–14306.

Lugaresi, G. and A. Matta. 2021. "Automated Manufacturing System Discovery and Digital Twin Generation". *Journal of Manufacturing Systems* 59:51–66.

Lv, Y. Q., C. K. M. Lee, Z. Wu, H. K. Chan and W. H. Ip. 2013. "Priority-Based Distributed Manufacturing Process Modeling via Hierarchical Timed Color Petri Net". *IEEE Transactions on Industrial Informatics* 9(4):1836–1846.

Manolakis, D., N. Bosowski, and V. K. Ingle. 2019. "Count Time-Series Analysis: A Signal Processing Perspective". *IEEE Signal Processing Magazine* 36(3):64–81.

Milde, M. and G. Reinhart. 2019. "Automated Model Development and Parametrization of Material Flow Simulations". In *2019 Winter Simulation Conference (WSC)*, 2166–2177 https://doi.org/10.1109/WSC40007.2019.9004702.

Nikolenko, S. I. 2021. *Synthetic Data for Deep Learning*, Volume 174. Springer International Publishing.

Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan *et al*. 2019. "Pytorch: An Imperative Style, High-Performance Deep Learning Library". *Advances in Neural Information Processing Systems* 32:8026–8037.

Paulin, G. and M. Ivasic-Kos. 2023. "Review and Analysis of Synthetic Dataset Generation Methods and Techniques for Application in Computer Vision". *Artificial Intelligence Review* 56(9):9221–9265.

Phanden, R. K., P. Sharma, and A. Dubey. 2021. "A Review on Simulation in Digital Twin for Aerospace, Manufacturing and Robotics". *Materials Today: Proceedings* 38:174–178.

Reddy, G. B., S. S. Murty, and K. Ghosh. 1993. "Timed Petri Net: An Expeditious Tool for Modelling and Analysis of Manufacturing Systems". *Mathematical and Computer Modelling* 18(9):17–30.

Rocha, E. M. and M. J. Lopes. 2022. "Bottleneck Prediction and Data-Driven Discrete-Event Simulation for a Balanced Manufacturing Line". *Procedia Computer Science* 200:1145–1154.

Saez, M., F. P. Maturana, K. Barton, and D. M. Tilbury. 2018. "Real-Time Manufacturing Machine and System Performance Monitoring Using Internet of Things". *IEEE Transactions on Automation Science and Engineering* 15(4):1735–1748.

Shamshad, A., M. Bawadi, W. Wanhussin, T. Majid and S. Sanusi. 2005. "First and Second Order Markov Chain Models for Synthetic Generation of Wind Speed Time Series". *Energy* 30(5):693–708.

Thiyagalingam, J., M. Shankar, G. Fox, and T. Hey. 2022. "Scientific Machine Learning Benchmarks". *Nature Reviews Physics* 4(6):413–420.

Wang, G. C. and K. Gross. 2018. "Real Time Empirical Synchronization of IoT Signals for Improved AI Prognostics". In *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, 954–959: IEEE.

Yin, J., W. Rao, M. Yuan, J. Zeng, K. Zhao, C. Zhang *et al*. 2019. "Experimental Study of Multivariate Time Series Forecasting Models". In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2833–2839. New York, NY, USA: ACM.

AUTHOR BIOGRAPHIES

SILVAN MARTI is a PhD student in the Production Systems division of the Industrial and Materials Science Department at Chalmers University of Technology in Gothenburg, Sweden. His primary research interest is in the area of deep learning based multivariate time-series analytics in operations research for an industrial setting. His email address is silvan@chalmers.se and his website is https://research.chalmers.se/en/person/silvan.

PAULO VICTOR LOPES is a PhD student in the Operations Research Program at Aeronautical Institute of Technology and Federal University of São Paulo. His research interests include data driven modelling of digital twins, what-if experiments design and data-driven techniques to improve production lines performance. He is currently a guest researcher at the Industrial and Material Science Department of Chalmers University of Technology, Sweden. His email address is paulov@chalmers.se and his website is https://research.chalmers.se/en/person/paulov.

SIYUAN CHEN is a PhD student in Industrial and Material Science Department of Chalmers University of Technology. His research focuses on digital twins, smart maintenance and deep reinforcement learning. He is dedicated to developing an Artificial Intelligence tools enhanced data-driven Digital Twin model to support decision-making and analysis, and an integrated manufacturing analytical platform. His email address is siyuan.chen@chalmers.se and his website is https://research.chalmers.se/en/person/siyuanc.

MOHAN RAJASHEKARAPPA is a PhD student in the Production Systems division of the Industrial and Materials Science Department at Chalmers University of Technology in Gothenburg, Sweden. His research focuses on applying Industrial AI to improve smart maintenance systems. By integrating AI techniques, he aims to enhance the reliability and efficiency of industrial operations. His commitment to practical and innovative solutions in maintenance technology is central to his work, as he seeks to bridge the gap between academic research and industry application. His email address is rmohan@chalmers.se and his website is https://www.chalmers.se/personer/rmohan/.

ELHAM REKABI BANA is a PhD student in the Product Development division of the Industrial and Materials Science Department at Chalmers University of Technology in Gothenburg, Sweden. Her research focuses on developing and applying graph-based deep generative models for solving high-dimensional inverse problems within the realm of physical and production systems. Her email address is rekabi@chalmers.se and her website is https://research.chalmers.se/en/person/rekabi.

AMON GÖPPERT is a postdoc and chief engineer at the Chair of Intelligence in Quality Sensing (IQS) in the Laboratory for Machine Tools and Production Engineering (WZL) of RWTH Aachen University. At the WZL he is leading the department of sensing and robotics with the research groups of adaptive cognitive robotics and dynamic systems planning. In the latter, his research on production control lies in the integration of simulation modelling with machine learning algorithms. His email address is amon.goeppert@wzl-iqs.rwth-aachen.de and his website is https://www.wzl.rwth-aachen.de/go/id/siic/gguid/0x68E6AA030C35D246AA48C9E74377B882.

MÉLANIE DÉSPEISSE is an Associate Professor in Sustainable Digitalized Production at Chalmers University of Technology in Gothenburg, Sweden. Her research focuses on the relationship between industry and environmental sustainability at various levels, from resource efficient manufacturing operations to technology roadmapping. Her research aims to develop tools and methods to help companies transition towards more eco-efficient and circular industrial systems. Her email address is melanie.despeisse@chalmers.se and her website is https://research.chalmers.se/en/person/despeiss.

JOHAN STAHRE is a professor at Industrial and Materials Science, Chalmers University of Technology in Gothenburg, Sweden. He leads the division of Production Systems. His research is on production automation and human factors. He has coordinated national and international projects on social sustainability, levels of automation and collaboration, and human supervisory control of production. His email address is johan.stahre@chalmers.se and his website is https://research.chalmers.se/person/jost.

BJÖRN JOHANSSON is a professor in Sustainable Production and Vice Head of Production Systems division at the Department of Industrial and Materials Science, Chalmers University of Technology in Gothenburg, Sweden. He serves as Production Modeling Corporation director for the European office in Gothenburg. His research focuses on the area of Discrete Event Simulation applied for manufacturing industries, including environmental effects modeling, modular modeling methodologies, software development, user interfaces, and input data architectures. His email address is bjorn.johansson@chalmers.se and his website is https://research.chalmers.se/en/person/job.