Leveraging Synthetic Data and Machine Learning for Shared Facility Scheduling

Marsel Rabaev[1], Handy Pratama[1], and Ka C. Chan[2]

[1] The University of New South Wales, Sydney, Australia
[2] University of Southern Queensland, Toowoomba, Australia
Kc.chan@usq.edu.au

Abstract. This research explores the applicability of machine learning (ML) algorithms in addressing key challenges in manufacturing planning and control (MPC), with a specific focus on capacity requirement planning (CRP) and scheduling. To effectively train ML algorithms, a discrete-event simulation (DES) methodology is employed to construct a system model, generating synthetic data through simulations across diverse scenarios. The proposed framework's efficacy is empirically evaluated through three distinct case studies, involving sequential, parallel, and shared facility layouts. The sequential and parallel layouts assess overall feasibility and capacity requirements planning, while the shared facility layout investigates scheduling within a more complex flexible manufacturing system. The research findings provide compelling evidence supporting the utilization of synthetic data for training ML models, facilitating efficient resolution of facility scheduling challenges in manufacturing.

Keywords: Manufacturing planning and control, discrete-event simulation, machine learning, facility scheduling, system modeling, synthetic data

1 Introduction

The utilization of ML approaches has gained significant traction in solving diverse manufacturing and supply chain problems, consistently delivering satisfactory results. However, a major obstacle in the application of ML techniques to manufacturing lies in the requirement for substantial amounts of training data, which is often scarce or not readily available. To address this challenge, computer simulation offers a viable solution by generating synthetic data that can be employed to train ML models effectively. Particularly, DES has been extensively utilized in solving shop floor problems, making it a promising gateway for synergizing simulation and machine learning techniques. By leveraging DES as a means to generate synthetic data, the fusion of simulation and ML holds potential for advancing manufacturing research and implementation.

2 Literature Review

Dogan & Birant (2021) classified ML-related studies in industrial engineering into five areas: quality control, condition and operations monitoring, demand forecasting, and production scheduling. While Kang et al. 2020 literature review suggests that ML adoption is widely researched in quality control and condition monitoring, Fahle et al. 2020 review revealed a lack of research papers on assistance systems and learning factory training concepts. As a result, this review focuses on these latter two areas.

Compared to traditional methods, Bajari et al. (2015) demonstrated that supervised ML models produce more accurate forecasts. Furthermore, Dou et al. (2021) findings suggest that deep learning outperforms other approaches, and incorporating multiple factors improves forecast accuracy. Additionally, Wiyanti et al. (2021) found that deep learning shows promising results compared to several other models using RMSE as a performance metric.

One major obstacle to implementing machine learning techniques in manufacturing is the requirement for massive amounts of data to train the ML model (Wuest et al., 2016), which is typically not widely available. However, the discrete event simulation technique can generate synthetic data that can be used to train an ML learning model. For example, Pfeiffer et al. (2016) used a DES model to generate synthetic data and used an ML framework to select tuning parameters that improve accuracy and robustness for multi-model based prediction of manufacturing lead times in the scheduling field. Similarly, Silva et al. (2017) examined the use of artificial neural networks to predict capacity in simulated supply chains in the field of supply chain management, while Gyulai et al. (2014) used ML to determine cost-efficient configurations of reconfigurable production lines across various products. However, the application of ML with DES-generated data in MPC framework has yet to be thoroughly studied.

Previously, the authors developed a generic framework that generates synthetic data via DES (Chan et al., 2022), resulting in three publicly available datasets corresponding to different manufacturing layouts: sequential, parallel, and shared facility layouts. This paper expands beyond data generation and focuses on leveraging the synthetic data to train ML models for facility scheduling. The primary objective is to evaluate the performance of these trained ML models. The effectiveness of the proposed framework will be assessed by measuring the accuracy achieved by the trained ML models, while efficiency will be evaluated in terms of the time required for data generation and training processes.

3 Methodology

This research employs a systematic approach to achieve its objective of developing a comprehensive framework that facilitates synthetic data generation, machine learning

(ML) model training, and evaluation. As shown in Figure 1, the proposed methodology includes a series of steps, which will be further discussed below.

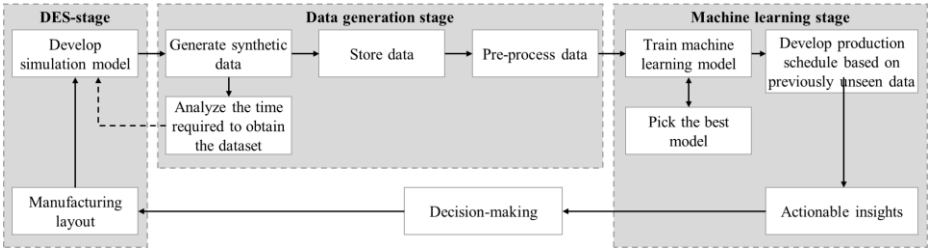

Figure 1. Proposed Research Methodology

The research methodology begins with the initial step of defining the layout and demand behavior of a manufacturing system, which helps determine the complexity of the manufacturing scheduling problem. Three system layouts from simple to complex were designed and experimented with in this study. Based on the defined manufacturing layout, a DES model that mimics the behavior and dynamics of the actual manufacturing system is built using ARENA software. The simulation models are then utilized to generate manufacturing process data, leveraging the advantages of DES in providing clear and usable data by eliminating process interruptions and external interventions. This makes the data suitable for feature selection and the application of machine learning (ML) techniques for schedule generation.

To train the ML models, MATLAB's Neural Network fitting and Regression Learner tools are employed, specifically designed for predicting continuous target values. In the case of the first and second simulation layouts, the Neural Network fitting tool is used to investigate overall feasibility, focusing on predicting process utilization. However, this paper skips the analysis of the first two layouts and focuses on the more complex flexible manufacturing system. For the last layout, we explored five main groups of ML algorithms, resulting in a total of 16 ML models that were empirically tested. The five groups, ranging from simple linear regression to cubic Support Vector Machines (SVM) with the aim of evaluating the performance of various ML algorithms using the same synthetic dataset. The algorithm exhibiting the best performance in terms of both Root Mean Square Error (RMSE) and training time is selected for further development of a production schedule.

The machine learning algorithms and sub-types evaluated include:

- Artificial Neural Network: MATLAB Neural Network Tool
- Support Vector Machine (SVM): Linear SVM, Quadratic SVM, Cubic SVM, Coarse Gaussian SVM, Medium Gaussian SVM, Fine Gaussian SVM
- Regression Tree: Coarse Tree, Medium Tree, Fine Tree
- Ensemble Tree: Boosted Tree, Bagged Tree
- Linear Regression: Regular Linear Regression, Stepwise Linear Regression, Robust Linear Regression, Interaction Linear Regression

The performance of the trained models is assessed based on two criteria: RMSE and training time. These metrics serve as indicators of the models' accuracy and efficiency. The resulting ML models are then employed to predict target values using previously unseen data, and the obtained results are cross validated with the simulation data using inferential statistical tests. Finally, the MATLAB code representing the developed ML models is compiled and saved for future use. This code can predict process utilization and generate a production schedule for manufacturing processes.

The proposed framework is highly versatile, making it suitable for implementation across various layouts of manufacturing facilities. Moreover, it exhibits scalability, empowering it to effectively handle complex manufacturing processes that involve numerous internal and external production variables. However, due to the differences in features and target values among layouts, a separate ML model must be trained for each specific case.

While this approach offers several advantages, there are also limitations to consider. For instance, obtaining live data from industrial companies typically results in unstructured data with numerous outliers and missing values, necessitating rigorous pre-processing to ensure accuracy. Moreover, the reliance on licensed software, such as MATLAB, during the machine learning (ML) process restricts the seamless integration of the resulting model code with ERP software applications. However, ML models can be implemented in programming languages like Python, utilizing libraries such as Scikit-learn and TensorFlow, allowing for seamless integration of the ML process into business operations. Overall, this approach provides a time-saving technique for testing hypotheses regarding the effectiveness of ML in generating schedules within the domains of MPC and SCM.

4 Case Study – Shared Facility Scheduling

The flexible manufacturing system layout, initially proposed by Slack et al. (2013) and utilized in the authors' previous study on synthetic data generation using DES (Chan et al., 2022), incorporates increased complexity in the manufacturing process. The manufacturing site produces four distinct SKUs (Stock Keeping Unit), each with unique parameters including demand, weight, type, and processing time at different stations. The layout is shown in Figure 2 below.

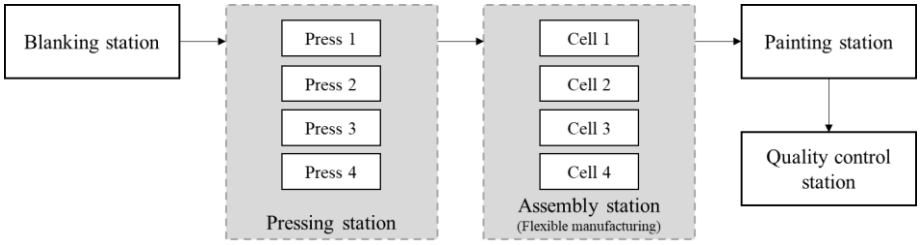

Figure 2. Flexible Manufacturing System Layout

A forklift is used to transport parts in batches between stations. The batch size is determined by the total weight and a maximum carrying capacity of two tonnes. Each part is assigned a specific weight at the start of the simulation, with four constant values corresponding to different SKU types, as shown in Table 1. To simulate the forklift operations, distance values between stations are incorporated in ARENA, along with stochastic travel times. Loading and unloading times at each station remain constant, but there may still be instances of part queuing due to the assumption of infinite-sized storage.

Table 1. SKU Parameters

Part type	Weight (kg)	Cell route	Processing time (seconds)	
			Mean (μ)	Std. Dev (σ)
SKU1	1.5	Cell 1	25	0.1
SKU2	2.5	Cell 1, Cell 2, Cell 3	15	0.1
SKU3	4.0	Cell 3, Cell 4	23	0.1
SKU4	4.5	Cell 1, Cell 2, Cell 4	17	0.1

The model starts with the uncoiled coils, used to produce blanks at the blanking station. These blanks are then loaded onto a forklift and transported to the pressing station, where they are unloaded and assigned to the appropriate press based on their SKU type. Following the pressing process, the parts are sent to the assembly station, which comprises four cells dedicated to processing different SKUs along predetermined routes. The selection of cells is determined primarily by cell availability. If a required cell is not available, the part awaits until the designated cell becomes free. The processing times for each SKU are specified in Table 1.

The final station, paint and quality, simulates two conveyors with a capacity of 3,600 parts each, operating simultaneously. Additionally, a quality control station is included in the process (Harun & Cheng, 2012). The paint conveyors encompass multiple processes, such as primer coating, painting, and furnace drying. In this simulation model, a significant volume of synthetic data was generated to capture and store crucial raw data without omitting any key information that may be needed in the future ML stage.

5 Experimental Results

5.1 Feature Selection and Machine Learning Model Training

The simulation model generated numerous features, and improving training efficiency and model performance relies on selecting the right set of key features and excluding redundant features. For example, in the shared facility, the ML model only requires one but not both of the highly correlated features, such as station utilization or waiting time

for predictions. However, essential information including total finished parts per SKU and specific SKU combinations processed by assembly cells is necessary. Pearson's correlation analysis of the reduced feature set demonstrated a strong positive correlation between the total number of finished parts and their respective assembly cells (Table 2).

Table 2. Pearson's correlation for the last layout

Target values	Predictors				Target values	Predictors			
	SKU1	SKU2	SKU3	SKU4		SKU1	SKU2	SKU3	SKU4
Cell1					Cell4				
SKU2	-0.348	0.992	-0.172	-0.311	SKU3	-0.282	-0.275	1.000	-0.311
Cell2					Cell1				
SKU2	-0.338	0.992	-0.185	-0.321	SKU4	-0.782	-0.225	0.370	0.626
Cell3					Cell2				
SKU2	0.048	0.646	-0.793	0.102	SKU4	0.354	-0.644	0.035	0.576
Cell3					Cell4				
SKU3	-0.283	-0.282	1.000	-0.305	SKU4	0.276	0.103	-0.880	0.573

In our experiments, we studied various algorithms while focusing on a single target value – the number of SKU2 processed by Cell1 (Cell1 SKU2). The training results of the machine learning models are provided in Table 3.

Table 3. Summary of the machine learning models' performance

Algorithm	RMSE	Training time (sec)	Algorithm	RMSE	Training time (sec)
ANN	81.271	47.00	Medium Tree	85.654	51.21
Linear Regression	82.465	28.26	Coarse Tree	84.895	45.93
Interaction Linear	81.351	33.85	Linear SVM	n/a	n/a
Robust Linear	82.466	33.01	Quadratic SVM	81.340	91858.00
Stepwise Linear	82.465	31.05	Cubic SVM	n/a	n/a
Ensemble Boosted Tree	478.75	276.02	Fine Gaussian SVM	n/a	n/a
Ensemble Bagged Tree	84.366	430.93	Medium Gauss SVM	81.799	143140.0
Fine Tree	86.309	66.52	Coarse Gaussian SVM	81.432	164000.00

The study found that using ANN produced the best performance, with an RMSE of 81.27, although it took almost two times longer to complete compared to linear regression. The RMSE means that predicting values for Cell1 SKU2 within a range of 7446 to 15306 units per day will result in an error of around 0.74% of the mean value, which is very accurate. It is noteworthy that the algorithm's exceptional performance can be attributed to its flexibility and capability in capturing complex relationships within manufacturing data. This stems from its layered structure and the incorporation of numerous parameters. This flexibility allows ANNs to fit a wide range of data patterns, which proves advantageous in scenarios where relationships are intricate and not easily captured by simpler algorithms such as Linear Regression or Tree-based methods.

Some training sessions for linear, cubic, and fine Gaussian SVMs were interrupted after exceeding 100 hours due to technical and time constraints. It was also observed that longer training times did not necessarily improve model performance.

Ease of use and greater flexibility provided by the Neural Network fitting has predetermined which algorithm has the potential to be used further. In Table 4 are summarized results of ANN application. The parameters of the ANN are similar to those given in the Table 3: the number of predictors is four (the total number of finished SKU1, SKU2, SKU3, SKU4), the number of target values is one (eight models have been trained) and the number of samples is 605620.

Table 4. ANN training results for the eight target values

Target value		ML model performance		
SKU type	Cell index	RMSE	% error of the mean value	Training time, (sec)
SKU2	Cell1	81.301	0.743%	69
	Cell2	23.366	0.585%	143
	Cell3	100.498	3.504%	121
SKU3	Cell3	6.308	0.138%	101
	Cell4	6.316	0.575%	24
SKU4	Cell1	301.662	7.065%	7
	Cell2	163.401	1.538%	22
	Cell4	305.941	4.854%	70

The ANN has performed well in terms of RMSE and training time. For instance, Cell3 SKU3 achieved an RMSE of approximately 6, which is only 0.13% of the mean error. However, accuracy varies across SKUs.

Table 5. Normality test results and 95% tolerance interval

Data source	Normality test, p-value	T-Test's p-value	Data source	Normality test, p-value	T-Test's p-value
Cell1 SKU2 original	0.126	0.969	Cell4 SKU3 original	0.008	0.996
Cell1 SKU2 predicted	0.202		Cell4 SKU3 predicted	0.008	
Cell2 SKU2 original	0.135	0.925	Cell1 SKU4 original	0.079	0.953
Cell2 SKU2 predicted	0.369		Cell1 SKU4 predicted	0.121	
Cell3 SKU2 original	0.815	0.703	Cell2 SKU4 original	0.799	0.906
Cell3 SKU2 predicted	0.844		Cell2 SKU4 predicted	0.027	
Cell3 SKU3 original	0.011	0.989	Cell4 SKU4 original	0.319	0.919
Cell3 SKU3 predicted	0.010		Cell4 SKU4 predicted	0.021	

The resulting machine learning models were utilized to predict previously unseen data from the 3000-sample dataset used to study time dependency. The predicted schedule was cross-validated using a t-test. All values except for the number of SKU2 processed by Cell3, both original and predicted, followed a Gaussian distribution. Therefore, the central limit theorem was applied to Cell3 SKU2. The results of the normality test in Table 5 indicate that the samples are normally distributed, allowing for inferential statistical tests to be employed.

The T-Test has been employed; the null hypothesis has been established in the following way: the mean of the original dataset is similar to the mean of the predicted dataset. The p-values in Table 5 show that there is not enough evidence to reject the null hypothesis. Hence, referring to the initial primary hypothesis of the research it can be concluded that the ANN is capable of creating shared facility scheduling based on the demand values.

6 Discussions and Conclusion

The machine learning models that were trained have successfully completed the tasks assigned to them. The algorithms used in machine learning have demonstrated high reliability in making predictions. During the case study, errors ranged between 0.14% to 7% of the mean number of processed SKUs by a specific cell. In most cases, the time required to complete the training process was between 0 and 120 seconds. The results were confirmed through a t-test, which proved the primary hypothesis that machine learning can be applied for manufacturing planning and control tasks, specifically capacity requirements planning and finite scheduling. The importance of feature selection was demonstrated, and the correlation between target values and predictors was applied to corresponding performance of the machine learning model, showing the relationship between the accuracy of predictions and coefficients of correlation. The higher the degree of correlation between target values and predictors, the more accurate the result. While adopting the ANN algorithm, we recommend considering several factors:

- Model Complexity and Flexibility: Artificial Neural Networks are highly flexible and capable of capturing complex relationships in data due to their layered structure and numerous parameters. This flexibility allows ANNs to fit a wide range of data patterns.
- Feature Learning: ANNs can automatically learn relevant features from data, reducing the need for manual feature engineering. This is especially useful when dealing with high-dimensional or unstructured data, as ANNs excel in extracting meaningful representations.
- Non-linearity: Many real-world problems exhibit non-linear relationships among variables. ANNs, with their activation functions and hidden layers, can effectively model these non-linearities, contributing to their superior performance over linear methods like Linear Regression.

- Computational Resources and Training Time: Training times required for different algorithms vary significantly. While ANNs may have outperformed other algorithms, they often require longer training periods and greater computational resources. This trade-off between performance and training time/resources should be thoughtfully considered in practical applications.

While ANNs have shown impressive performance in these research cases, they may not be the best choice for all situations. Factors such as the nature of the problem, dataset size, resource availability, interpretability requirements, and others can influence the suitability of alternative algorithms. Comprehensive experimentation and exploration of different algorithms are advised to attain optimal results, as demonstrated in this research.

The results demonstrated throughout the research showed that the proposed framework is feasible and its implementation can be further studied at an industrial project. It has been demonstrated that the utilization of DES to generate synthetic data is a fast and reliable approach, which can be used in the initial stages of machine learning algorithms implementation to perform MPC tasks in the industrial stage, particularly to replace time consuming data collection at the shop floor. Moreover, the machine learning algorithms has also demonstrated high degree of reliability of the results, providing rapid support in decision making process.

References

Bajari, P., Nekipelov, D., Ryan, S. P., & Yang, M. (2015). Machine Learning Methods for Demand Estimation. *American Economic Review: Papers & Proceedings*, *105*(5), 481–485. https://doi.org/10.1257/aer.p20151021

Chan, K. C., Rabaev, M., & Pratama, H. (2022). Generation of synthetic manufacturing datasets for machine learning using discrete-event simulation. *Production and Manufacturing Research, 10(1),* 337-353. https://doi.org/10.1080/21693277.2022.2086642

Dogan, A., & Birant, D. (2021). Machine learning and data mining in manufacturing. *Expert Systems with Applications*, *166*, 114060. https://doi.org/10.1016/J.ESWA.2020.114060

Dou, Z., Sun, Y., Zhang, Y., Wang, T., Wu, C., & Fan, S. (2021). Regional Manufacturing Industry Demand Forecasting: A Deep Learning Approach. *Applied Sciences 2021, Vol. 11, Page 6199*, *11*(13), 6199. https://doi.org/10.3390/APP11136199

Fahle, S., Prinz, C., & Kuhlenkötter, B. (2020). Systematic review on machine learning (ML) methods for manufacturing processes – Identifying artificial intelligence (AI) methods for field application. *Procedia CIRP*, *93*, 413–418. https://doi.org/10.1016/J.PROCIR.2020.04.109

Gyulai, D., Kádár, B., & Monostori, L. (2014). Capacity planning and resource allocation in assembly systems consisting of dedicated and reconfigurable lines. *Procedia CIRP*, *25*(C), 185–191. https://doi.org/10.1016/j.procir.2014.10.028

Harun, K., & Cheng, K. (2012). An integrated modeling method for assessment of quality systems applied to aerospace manufacturing supply chains. *Journal of Intelligent Manufacturing, 23*, 1365–1378. https://doi.org/10.1007/s10845-010-0447-7

Kang, Z., Catal, C., & Tekinerdogan, B. (2020). Machine learning applications in production lines: A systematic literature review. *Computers & Industrial Engineering*, *149*, 106773. https://doi.org/10.1016/J.CIE.2020.106773

Pfeiffer, A., Gyulai, D., Kádár, B., & Monostori, L. (2016). Manufacturing Lead Time Estimation with the Combination of Simulation and Statistical Learning Methods. *Procedia CIRP*, *41*, 75–80. https://doi.org/10.1016/j.procir.2015.12.018

Silva, N., Ferreira, L. M. D. F., Silva, C., Magalhães, V., & Neto, P. (2017). Improving Supply Chain Visibility with Artificial Neural Networks. *Procedia Manufacturing*, *11*, 2083–2090. https://doi.org/10.1016/j.promfg.2017.07.329

Slack, N., Chambers, S., & Johnston, R. (2013). Operations Management. In *Operations Management*. https://doi.org/9780132342711

Wiyanti, D. T., Kharisudin, I., Setiawan, A. B., & Nugroho, A. K. (2021). Machine-learning algorithm for demand forecasting problem. *Journal of Physics: Conference Series*, *1918*(4), 042012. https://doi.org/10.1088/1742-6596/1918/4/042012

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