Proceedings of the 2025 Winter Simulation Conference
E. Azar, A. Djanatliev, A. Harper, C. Kogler, V. Ramamohan, A. Anagnostou, and S. J. E. Taylor, eds.

USING DEEP LEARNING TO IMPROVE SIMULATION-BASED DECISION MAKING BY PROCESS LEAD TIME PREDICTIONS

Christian Schwede[1,2], and Adrian Freiter[2]

[1]Dept. of Engineering and Mathematics, University of Applied Sciences and Arts Bielefeld, Bielefeld, GERMANY
[2]Dept. of Industrial Engineering, Fraunhofer Institute of Software and Systems Engineering, Dortmund, GERMANY

ABSTRACT

Based on digital twins, simulation is often used in companies for the regular planning and control of operational processes. However, when modeling the lead times of individual processes, mean values of process times measured in advance are often used, which can lead to errors in the planning. This work demonstrates how models of these time distributions can be created and updated within a digital twin framework using machine learning. The lead times are used in simulation to create schedules. The approach is validated using the online order workforce scheduling of a medium-sized company that assembles individual packages of office materials for its customers.

1 INTRODUCTION

Decision support systems based on digital models such as digital shadows or digital twins (DTs) that represent the current state of the shop floor are used more and more for production planning and control (PPC) in companies (Kritzinger et al. 2018, Kuehner et al. 2021). Discrete-event simulation (DES) is often used in such systems for three use cases (Pfeiffer et al. 2015): *offline validation* of production plans during production planning, *online anticipatory recognition* of future deviations, and *online evaluation of short-term adaptations* during production control.

An important basis for simulation is the lead time of individual processes on the shop floor. The *process lead time* of a single process must be distinguished from *order lead time* which describes the overall lead time of the whole order. The process lead times influence, e.g., the occupation of resources, and the reliability of the delivery dates. Especially in the case of short-term production control, accurate lead times are vital. While production planning could be performed with the aim of a robust schedule based on various possible future scenarios, the short-term character of production control makes it necessary to focus on the most probable scenario. Nevertheless, often little effort is put into an accurate lead time prediction. The most approaches use unconditional probability distributions for the lead time prediction (Schwede and Fischer 2024). Even though process lead times especially in manual processes vary and are affected by a lot of conditions (e.g., order complexity, staff abilities, accessibility of material, day of the week, etc.) still offline estimation of uniform or normal distributions is the standard method. As shown in the literature review of Schwede and Fischer (2024), adding of conditions can improve the accuracy of the statements of simulation based digital twins that are made.

Considering more complex lead time estimations some work has been done mainly in two areas: 1. Order lead time estimation via machine learning (ML) has been used as an alternative to simulation-based prediction in combination with optimization methods (Lingitz et al. 2018) or to evaluate deviations found via online-simulation during production control (Gyulai et al. 2018b). 2. Process lead times have been estimated using ML to improve production planning (Bender et al. 2022, Rizzuto et al. 2021, Müller and Grumbach 2023, Yamashiro and Nonaka 2021).

While the ML-based prediction of order lead times can be either used as a faster or additional method to DES, in complex environments the level of detail of simulation-based prediction cannot be reached. ML-based process lead time prediction on the other hand shows promising results and could be used to improve the results of optimization methods (Yamashiro and Nonaka 2021) as well as the accuracy of the DES.

In this work, we will investigate the effect of ML-based lead time prediction in the context of manual workplaces and a highly dynamic environment. Furthermore, we will present an approach to integrate lead time model training in a DT framework. The contribution of this work is the integration of lead-time prediction of manual work process into simulation-based digital twin. Especially we investigate the effect of the improved prediction of single lead-times on the overall simulation accuracy and the underlining optimization task. The application case is the online workforce scheduling of a medium-sized company that assembles individual packages of office materials for its customers. The process lead time influences the creation of the order sequence as well as the workforce schedule.

The rest of the work is structured as follows. In section 2 the relevant literature on lead time prediction is summarized. In section 3 the application case is described and section 4 presents experiments and results of using ML-based lead time prediction models. Section 5 summarizes the results with conclusions and presents an outlook to future work.

2 RELATED WORK

Lead time prediction can be divided into two main areas: *Order lead time prediction* aims at predicting the time between order release and product delivery while *process lead time prediction* aims at predicting the time that is needed to perform a single process step. To solve the problem of order lead time prediction two approaches can be found in the literature (Burggräf et al. 2020): the first approach is *indirect* and based on a detailed knowledge of the processes in the factory. Single process times including waiting times are combined with a schedule to calculate or simulate the order lead time. Predictions of process lead time can be helpful in this area. The second approach is *direct* and aims at a prediction of the order lead time by historical data using ML methods or based on specific heuristics (e.g., Little's law).

Direct lead time prediction has recently been based on ML methods using historical data to train a function of the lead time depending on a wide set of parameters. These approaches tend to outperform simple heuristics in performance and simulation-based prediction in speed. Especially because of the latter aspect they are used in combination with optimization algorithms and serve as an evaluation function (Lingitz et al. 2018) or as an additional measure to evaluate deviations found via online simulation during production control (Gyulai et al. 2018b).

For process lead time prediction two approaches can also be distinguished: Estimations of *well-known simple probability distributions* such as uniform, normal or other simple distributions and prediction of *unknown complex distributions* via ML methods. While the first approach is widely used because of its simplicity, it assumes that the hidden distributions of the data are known and simple. If this is not the case, which is the fact for most cases of lead times in industry environments (Yamashiro and Nonaka 2021), this first approach leads to poor results. As in order lead time prediction, ML-based approaches to predict unknown more complex distributions have recently been presented in a wide range to improve the accuracy of production planning and simulation (Bender et al. 2022, Rizzuto et al. 2021, Müller and Grumbach 2023, Yamashiro and Nonaka 2021). In addition, if the distribution is too complex to learn with only one model, the use of more than one model could help. Therefore, a separator could be used that divides the problem into smaller sub-problems. As example, Smith and Dickinson (2022) considered it too complex to learn the distribution for three different product types in one model. Consequently, they decided to divide the problem into smaller subproblems and learn each product type with one model.

Concerning the methods applied a wide range of ML approaches is used: The literature review of (Burggräf et al. 2020) stated that artificial neural networks (ANNs), linear or logistic regressions (LRs), decision trees, random forests (RFs), support vector machines, and k-nearest neighbors are used within the ML methods. Additionally, it could be stated that ANNs are used by far most frequently with 43 %, followed by LRs with 30 %, and RFs in third place with 22 %. If more complex data is used, such as product data, it is stated that only ANNs and RFs are used because these approaches are better at handling complex data inputs. Few works compare different approaches with each other (Burggräf et al. 2020). Next to the hands-on ML approaches, it is noticeable that automated ML (often referred to by the abbreviation AutoML) solutions are more frequently used (Bender and Ovtcharova 2021, Bender et al. 2022, Sousa et al. 2022). Even if there is still some work to be done to be regarded as a suitable alternative (Bender et al. 2022). For validation, the k-fold cross-validation method is used most frequently (Biazon de Oliveira et al. 2021, Sousa et al. 2022).

The transferability of models to other applications proves to be difficult because the input data and other control rules used are different, even though explainability and transfer learning become more important (Panigrahi et al. 2021). That is why a comparison with other approaches is only possible to a limited extent. Additionally, comparing two approaches that use unsuitable accuracy measures (Hoffmann et al. 2019) makes it difficult to benchmark with other applications. Common accuracy measures, such as Root Mean Squared Error (RMSE) (e.g., Müller and Grumbach 2023, Rizzuto et al. 2021, Lingitz et al. 2018) or Normalized RMSE (NRMSE) (Gyulai et al. 2018a, Lingitz et al. 2018) and R^2 (Pfeiffer et al. 2016) can be used.

3 APPLICATION CASE

The application case is an online order and workforce scheduling of a medium-sized company that assembles individual packages of office materials for its customers. Customer orders arrive constantly and have assigned priorities concerning the level of urgency. If material is not available, and this is known in advance, the affected orders will have to wait. The orders are aimed to be completed at the time of shipping which leads to the situation that an already planned order schedule must be adapted several times during the day. Delivery reliability is an important key performance indicator. The company uses eight picking stations that are connected by conveyor belts as can be seen in Figure 1.

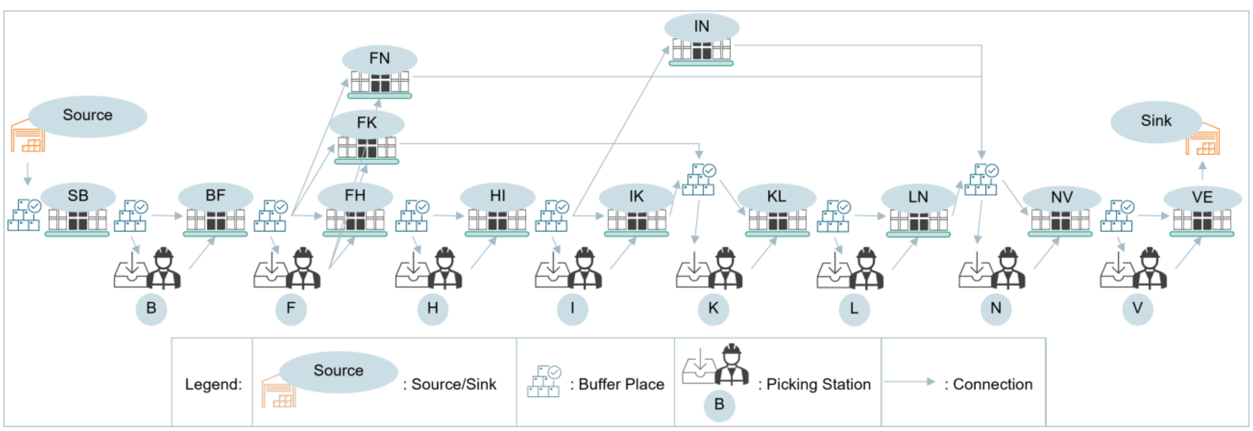

Figure 1:Material flow of the company connecting picking stations by conveyor belts.

The stations are situated inside the warehouse and picking is done by the workforce following a person-to-good principle taking out the parts needed from the shelves near each station. Every station has a fixed and disjoint subset of the 25,000 different picking materials assigned to it which are situated nearby. Orders are only entered into the system when sufficient material stock is identified in the IT system. Therefore, material availability is not considered a bottleneck in scheduling, although in real life it may happen that some parts are not available and have to be reordered, resulting in longer lead times and the need to reschedule.

A box is sent to the picking stations based on the materials listed in the picking order. Once at a picking station, the box is stored in an intermediate buffer waiting for its turn. The box should be selected by the picker based on the planned schedule, to ensure that urgent orders are delivered first. Based on a conventional picking system, the workforce manually picks up the goods from e.g., continuous racks and stores them in the boxes at the picking station. Since it is manual work, the process lead time varies depending on the materials to be picked because of different volumes, heights, distances and other material properties (part/material data). In addition, the assigned workers that react differently to the circumstances such as the last processes performed by the worker and the temperature today (workforce data and system status). Moreover, the process lead times depend on information about the order, e.g., urgency, the target quantity of the parts to be picked and the material already packed in the box is considered (order and position data). Furthermore, the company has a certification for the inclusion of workers with disabilities that may lead to further impact on the variance of process lead times. Once the box has passed all relevant stations, the order is completed, and the box is stored in the outbound buffer. Aside from these picking orders, there are also stocking orders to refill the storage places. The packages enter the station in the same way, the only difference is that they have to be unpacked to the shelves rather than packed with material from the shelves. Stocking is only performed in two of the stations.

As simulation, the multi-agent-based simulation (MAS) component of the Open Factory Twin (OFacT) framework is used (Schwede and Freiter 2024). The MAS allows the decentralized solution of the two sub-problems. On the one hand the sequence of picking and stocking orders to be released into the system has to be determined. This is solved by the order release agent. On the other hand, the order sequence for each picker has to be generated, which is executed by the worker resource agent. Since the number of pickers is smaller than the number of picking stations, the decision includes switching from one picking station to another. The overall goal is firstly to minimize the number of late order deliveries, secondly to minimize the sum of minutes of orders being late thirdly to minimize the sum of order lead times of all orders. The goals dominate each other in the way that the inferior goal is only taken into account for solution with equal results for the superior goal. Additionally, the maximization of capacity utilization is of interest. Since orders enter the system at any time and lead time deviation is high, the problem has to be solved as an online-optimization problem. The decision of which order enters the system next is updated after each order entry and the decision on which order to perform next is taken for each workforce when they have finished the last one.

To investigate the impact of a more precise process lead time prediction, fast local heuristics are designed based on expert knowledge. As input for both local heuristics, the remaining order lead times are determined. These are calculated as the sum of all remaining process lead times. As the transport times have a low variance, they are added as deterministic values based on the interim stops the order will take on the conveyor belt. In contrast, the remaining process lead times for picking and stocking are predicted.

$$
\begin{aligned}
&latest\_release\_date = remaining\_time\_to\_delivery - remaining\_order\_lead\_time \qquad (1)\\
&remaining\_time\_to\_delivery \ = \ planned\_delivery\_date - current\_time\\
&remaining\_order\_lead\_time \ = \ sum(remaining\_transport\_times) + \\
&sum(remaining\_process\_times)
\end{aligned}
$$

- The **order release sequence** is generated by sorting the orders due to their urgency. To calculate the latest release date (Eq. 1), the remaining time to delivery is subtracted by the determined remaining order lead time (which in this case is the total order lead time, since the orders have not been released). The orders are then sorted according to the latest release date in ascending order. Every time an order should be released to the system, the order with the earliest release date is chosen. To ensure that the system has a constant workload, only after an order leaves the system, another one is allowed to enter the system (continuous work in progress).
- The **work order sequence** for each picker is generated in the same manner. For all orders that can be performed by the picker (all orders in the buffer at the current and all free picking stations) the sequence of orders due to their urgency is calculated as described above. Only in this case solemnly the remaining processes are considered for the sum of the remaining order lead time. The most urgent orders are used for each work order assignment to an employee (function

assign_order_positions). In addition, to avoid station overflows, stations with buffers reaching a level near the capacity limit are prioritized. To prevent the workforce from switching constantly between the stations, the workforce will continue to process the orders until the intermediate buffer of the station is empty. But even if the picking station is not empty, the workforce can switch to another station if at least five processes are completed at the station and the current station is not prioritized (function check_b_allowed).

```
function assign_order_positions(aw, ab) returns a list of assigned order
positions
   Input: aw, currently available workers,
          ab, all buffers with a waiting order queue
   Local variables: sb, sorted buffers
                     b, a single buffer
                   uop, most urgent order process at a buffer
                   aop, assigned order positions

   # sort buffers: buffer >= limit first, rest sorted by most urgent
   orders
   sb = sort_buffer (ab)
   aop = []
   for b in sb: # most urgent first
      aw = []
      # get available workers and their distance to buffer
      for w in aw:
         w_allowed = check_b_allowed(w, b)
         if w_allowed:
            # determine the distance from w to b
            aw.append((w, get_distance(w, b)))

      # choose the next process based on the minimal latest release date
      uop = get_urgent_process(b)
      # choose the nearest worker
      aop.append((uop, argmin(aw[:, 1]))
   return aop

function check_b_allowed(w, br) returns bool value that states if the
worker can work at the buffer
   Input: w, worker,
         br, buffer requested
   Local variables: bc, current buffer of the worker

   if not has_necesarry_skill_br(w, br):
      return False
   bc = get_bc_w(w)
   if check_bc_queue_empty(bc, w):
      return True
   if check_w_executed_five_pos_at_bc(w) and not bc_reach_limit:
      return True
```

```
# skill to work at the current buffer
if not check_necessary_skill_bc(w, bc):
   return True
return False
```

It is ensured that two workers do not work at one picking station at the same time, which would lead to low-capacity utilization.

4 RESULTS

In this section the results are presented. In subsection 4.1 different methods to perform the process lead time prediction will be compared. The evaluation is based on the prediction loss of single process lead times of a test set. In subsection 4.2 the most promising prediction models are compared to the standard method of an estimation of a normal distribution. The models predict the lead time values used in the heuristics while for the evaluation in the simulation the real process lead times are applied (Figure 2). While only static data is used for the order release sequence due to unavailability of dynamics, they are considered in the next order heuristic. In this approach, it will be investigated to what extent the accuracy of the digital twin can be improved. As digital twin framework, the Open Factory Twin (OFacT) is used that allows a simple exchange of process time models and allows the usage of ML based process time models (Schwede and Freiter 2024). It is assumed that lead time predictions with higher precisions lead to a higher delivery reliability, since the order release sequence as well as the work order sequence are optimized based on the local heuristics described earlier. This is assumed since the urgency of each order can be predicted with a higher precision, meaning that more urgent orders can be prioritized in advance, leading to higher delivery reliabilities.

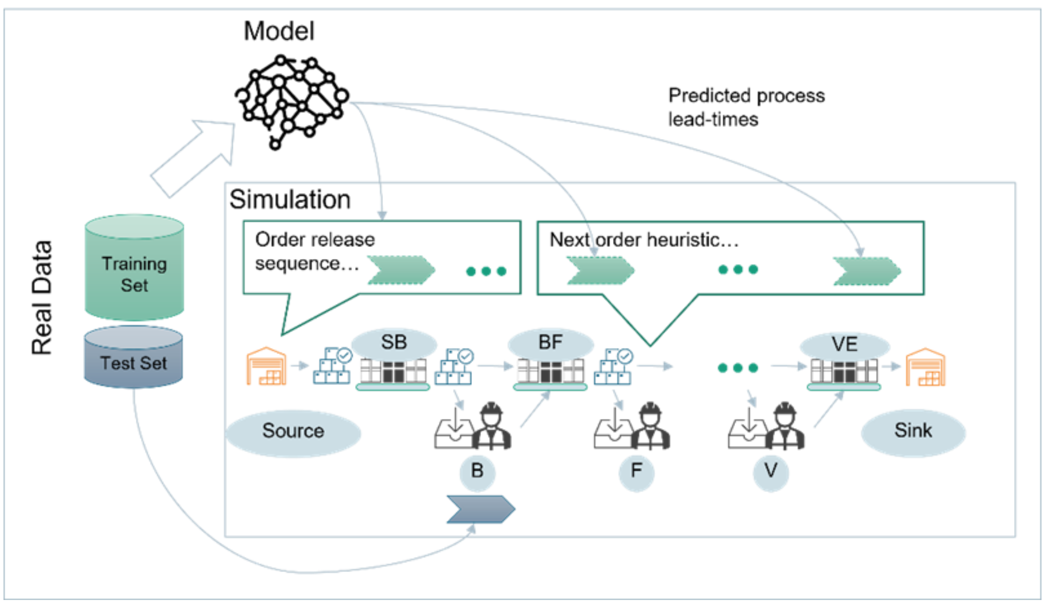

Figure 2: DT using the models to predict lead times.

From the 13 weeks of data available from the company, 12 weeks (training set) were used to train the lead time prediction models while the last week (test set) was used for evaluation.

4.1 Process Lead Time Prediction

To choose the right model for process lead time prediction the Cross Industry Standard Process for Data Mining (CRISP-DM) was applied (Chapman et al. 2000). The dataset consists of 13886 picking and 3592 stocking orders. Features were clustered into six groups (see Table 1).

Table 1: Feature Categorization.

Data Class	Feature from company dataset	Feature from DT
Order	2 of 23 (e.g., shipping method, urgent)	2 (order total volume so far, order total weight so far)
Process and Position	4 of 17 (e.g., target quantity, whole volume, whole weight)	-
Material	19 of 25 (e.g., product group name, SKU height, SKU quantity, distance)	-
Resource	1 of 2 (e.g., picking station)	-
Workforce	1 of 1 (e.g., workforce)	4 (e.g., last processes performed, last position of the same order)
System Status	4 of 4 (e.g., weekday, daytime, temperature today)	4 (e.g., work-in-progress, order releases (last five minutes), workers available)

To select relevant features, the importance values were derived from a RF trained with default values (sklearn v.1.3.0, with 100 estimator trees). An extract of the 15 features with the highest importance values that were included as input into the prediction model is depicted in Figure 3. With the importance values could be shown that “distance to storage place” has the highest importance on the process lead time followed by “target quantity”, “total volume”, and “total weight”. Also noticeable, features of all data classes can be found in the most important values.

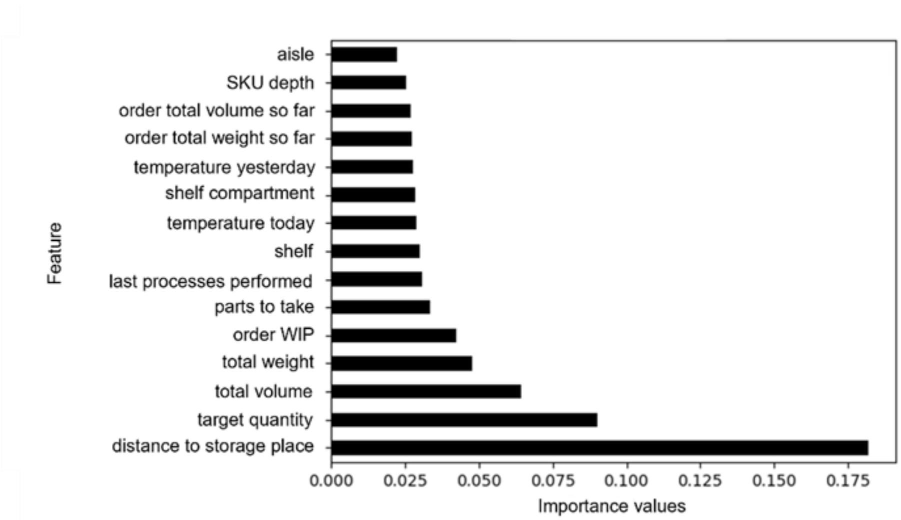

Figure 3: Importance Values (extract of the 15 highest values).

Based on the 82 features 41 (Stocking: 37) were selected for the prediction. From the company’s data 31 features were taken directly, and 10 were generated based on the DT. Since there are eight picking stations and two stocking stations the question arises of how many stations per prediction model are used (e.g., one per station, or one for all stations). To determine the number, the process lead time distributions of all processes were compared (Figure 4) to perform a preselection, and then the performance of the most promising combinations was measured.

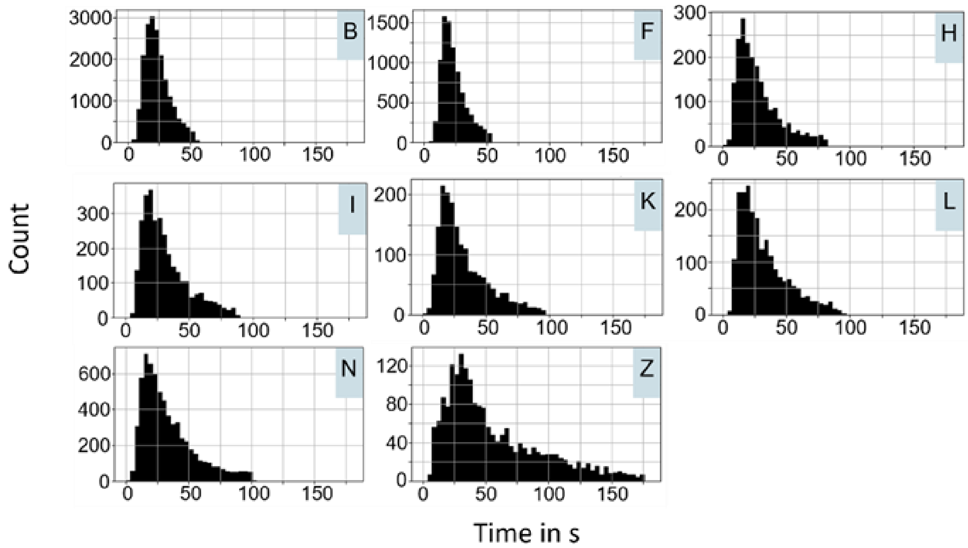

Figure 4: Lead time distribution per station.

We used four models covering the following processes: Picking BF, Picking HIKLN, Picking Z, and Stocking.

As ANNs were the models most used in literature, we decided to train four ANNs – one for each of the defined subproblems. The PyTorch library was used for the training and execution of the ANNs (Paszke et al. 2019). An exhaustive hyperparameter search was performed using Optuna (Akiba et al. 2019). The best-performing parameters with performance measure RMSE, 1,500 epochs, and k-fold cross-validation can be found in the Table 2. For the actual application, the epochs are increased to 3,000, leading to even better results.

Table 2: Hyperparameter Optimization.

Parameter name	Parameter value	Parameter name	Parameter value
Mini-batch size	{**32**, 64, 96, 128}	Optimizer	**Adam**, NAdam
Architecture type	{Pyramid, **Rectangle**}	Neurons per Layer	{16, 32, 48, 64, 96, 112, 128, 144, 160, 192, 224, 256, **512**}
Layers	{3, 4, **5**}	Learning rate	1e-5 - 1e-3 [**0,000169**]
Loss	{MSE, L1 (MAE), **Huber**}	Activation function	{**ReLU**, Mish, Softplus}
Batch Normalization	{**False**; True}	Dropout	{**False**; True}
Dropout (input)	0.4 – 0.9	Dropout (hidden)	0.2 – 0.6
Weight decay	{**False**; True}	Weight decay value	[1e-5, 1e-3]

Finally, we compared the standard normal distribution (ND) determined for each picking station separately with the ANN-based prediction (ML) on a dataset with and without numerical outlier (using an interquartile range of 1.5). The outliers are difficult to predict by the models since the specific circumstances are not covered by the data (e.g., stock out of materials, shirt term breaks). Based on the results represented in Table 3, the analysis of the RMSE reveals a decrease of about 19 % with numerical outliers and about 35 % without numerical outliers in comparison to the ND-based process lead time prediction. Looking at the behavior of the RMSE when including the numerical outliers, the error in the ML approach is more than twice as large. Especially for short-term planning and control, where larger relative deviations cannot be compensated, the results are promising.

Table 3: Model Evaluation with Test Dataset.

Measure Type	Model	Without numerical outliers	With numerical outliers
RMSE	ND	19.473	34.694
RMSE	ML	12.524	28.204
NRMSE	ND	1.712	2.183
NRMSE	ML	0.812	1.224

To get a visual insight into the improvements, the process lead times of 50 positions (extract of the dataset) are plotted in Figure 5. The green dots represent the real process lead times as a reference base. Next to them, the red dots represent samples from the ND and the blue dots represent the predictions of the ML-based approach. As expected, the ML-based prediction approach shows significantly better consistency with the real process lead time values. In contrast, the samples from the ND approach look rather random. However, as outliers are excluded during training, they cannot be represented appropriately in both approaches. This is also seen as the essential reason why the positive deviations are higher.

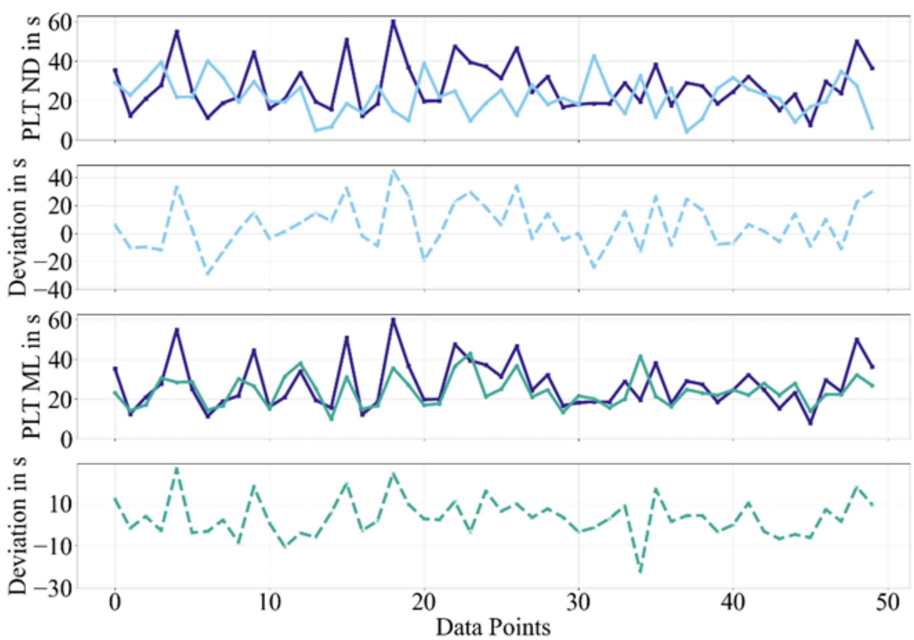

Figure 5: PLT Deviations from Real Values (a) ND Sampling and (b) ML-based Prediction.

4.2 Benefits of ML-based Prediction

In this subsection the effect of the more precise process lead time prediction on the online local heuristics optimization task. The indicators used for the comparison are average capacity utilization of the workforce (aCUW), average order lead time (aOLT), an average of remaining order lead time after end-of-production (aROLT), the sum of orders delayed (sOD), the sum of process lead time (sPLT), the sum of order positions processed (sPP) and the sum of orders processed (sOP). In the experiments, one week is simulated. Table 4 shows the results. The results show that the order delays were significantly reduced from 8.6 to 5 days, which is almost the half. This is further confirmed by the increase in aROLT. The increase in aROLT is measured instead of the sum of minutes of orders being late since the last system part (quality inspection and packing) is not considered in the simulation model but needs to be visited before the actual delivery. The aOLT could also be decreased slightly. In addition, the capacity utilization of the workforce remains equal, which also explains the equal sPLT, sPP, and sOP values.

Table 4: Results of ML-based prediction.

Model	aCUW [%]	sPLT [h:mm]	sPP	sOP	aOLT [h:mm]	aROLT [h:mm]	sOD [d:hh]
ND	88.23	37:55	4,085	1,188	6:35	11:51	8:14
ML	88.18	38:00	4,099	1,205	6:30	13:47	4:23

The results show that the ML-based lead time prediction improve optimization performed by the local heuristics. Since all of the mentioned goals of the optimization (minimize number of late order deliveries, sum of minutes of orders being late, sum of order lead times of all orders, and maximize capacity utilization) are directly or indirectly depended on the order lead time and the order lead time is used as main input parameter for the local heuristics, the increase of the input parameter precision lead to better output precision. Consequently, the higher precision of the order lead time in the planning phase leads to more optimal prioritization of the work orders. Therefore, the order lead times decreases as well as the delivery reliability increases slightly.

5 OUTLOOK AND DISCUSSION

In this work, we argued the importance of ML for the quality of planning and control in the context of DTs especially using simulation for decision support. We investigated this hypothesis in an industry case of using process lead time prediction for online scheduling of orders and workforce in the context of individual packaging of an office material supplier. The results are promising, first have shown that the ML based prediction has a 35 % smaller error without considering the outliers, but even with considering the outliers, the error could be reduced by 19 %. The better predictions can also lead to better accuracy of the simulation results especially considering the reduction of late customer orders almost 50 % using ML-based prediction instead of normal distributions (simple probability distribution).

In next steps different industry scenarios and more extensive time lines will be considered increase robustness of the results. In the scope of this work, we could only motivate the aspect of regularly updating the ML models within the DT with new data. Future research will focus on investigating the effects especially when major changes on the factory level have been applied in the real world. Here an interesting issue will be to constantly evaluate if the model still represents the real world data and if not to decide whether the model has to be updated or poor data qualities and outliers are the cause of the deviation. Especially outliers, as discussed above, can be tricky to handle. A possible approach would be to feedback the outliers to the company and eventually install further data capturing events to cover these exceptional situations and make them learnable for the models. Further work will also focus on extending the ML approach to the other process models to enhance prediction quality as well as to reduce manual modelling efforts. Another step will include learning the decision heuristics themselves, that are used on the shop floor from the data. These are capsulated within the simulation in agent behaviors which again facilitates the integration of ML approaches. There are two possible ways to learn the agent's behavior: The first one will be to train a classification model (e.g. a decision tree) with one class for each of the different possible decision to take. In this case the decision would be which order to process next (from the queue of the current work station or the one of the next stations). A challenging task here is to generalize the fixed number of classes in a way that the can represent the highly variable situations. Reinforcement Leaning is another way to learn an agent's behavior using the simulation as environment. Even tough here the goal would be to learn an optimal strategy rather than to copy the one applied by the workers.

REFERENCES

Akiba, T., S. Sano, T. Yanase, T. Ohta, and M. Koyama. 2019. "Optuna: A Next-generation Hyperparameter Optimization Framework". In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 4th-8th, Anchorage, USA, 2623-2631.

Bender, J. and J. Ovtcharova. 2021. "Prototyping Machine-Learning-Supported Lead Time Prediction Using AutoML". *Procedia Computer Science* 180: 649-655.

Bender, J., M. Trat, and J. Ovtcharova. 2022. "Benchmarking AutoML-Supported Lead Time Prediction". *Procedia Computer Science* 200:482-494.

Biazon de Oliveira, M., G. Zucchi, M. Lippi, D. Cordeiro, N. Da Rosa Silva, and M. Iori. 2021. "Lead Time Forecasting with Machine Learning Techniques for a Pharmaceutical Supply Chain". In *Proceedings of the 23rd International Conference on Enterprise Information Systems - Volume 1*, April 26th-28th, Online Streaming, 634–641.

Burggräf, P., J. Wagner, B. Koke, and F. Steinberg. 2020. "Approaches for the Prediction of Lead Times in an Engineer to Order Environment - A Systematic Review". *IEEE Access* 8:142434-142445.

Chapman P, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth. 2000. "CRISP-DM 1.0 Step-by-step Data Mining Guide". SPSS Inc.

Gyulai, D., A. Pfeiffer, G. Nick, V. Gallina, W. Sihn, and L. Monostori. 2018a. "Lead Time Prediction in a Flow-shop Environment with Analytical and Machine Learning Approaches". IFAC PapersOnLine 51(11):1029–1034.

Gyulai, D., A. Pfeiffer, J. Bergmann, and V. Gallina. 2018b. "Online Lead Time Prediction Supporting Situation-aware Production Control". *Procedia CIRP* 78: 190-195.

Hoffmann, F., T. Bertram, R. Mikut, M. Reischl, and O. Nelles. 2019. "Benchmarking in Classification and Regression". *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 9(5):e1318.

Kuehner, K. J., R. Scheer, and S. Strassburger. 2021. "Digital Twin: Finding Common Ground – A Meta-Review". *Procedia CIRP* 104:1227-1232.

Lingitz, L., V. Gallina, F. Ansari, D. Gyulai, A. Pfeiffer, W. Sihn, and L. Monostori. 2018. "Lead Time Prediction using Machine Learning Algorithms: A Case Study by a Semiconductor Manufacturer". *Procedia CIRP* 72:1051-1056.

Panigrahi, S., A. Nanda, and T. Swarnkar. 2021. "A Survey on Transfer Learning". In *Proceedings of ICICC 2019*. December 20th, Bhubaneswar, India, 781-789.

Müller, A. and F. Grumbach. 2023. "Predicting Processing Times in High Mix Low Volume Job Shops". In *Proceedings of the 16th International Doctoral Students Workshop on Logistics, Supply Chain and Production Management*, June 20th, Magdeburg, Deutschland, 89-95.

Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan et al. 2019. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, December 8th-14th, Vancouver, Canada, 8026-8037.

Pfeiffer, A., D. Gyulai, B. Kádár, and L. Monostori. 2016. "Manufacturing Lead Time Estimation with the Combination of Simulation and Statistical Learning Methods". *Procedia CIRP* 41:75-80.

Rizzuto, A., D. Govi, F. Schipani, and A. Lazzeri. 2021. "Lead Time Estimation of a Drilling Factory with Machine and Deep Learning Algorithms: A Case Study". *Proceedings of the 2nd International Conference on Innovative Intelligent Industrial Production and Logistics*, October 25th-27th, 84-92.

Schwede, C. and D. Fischer. 2024. "Learning Simulation-based Digital Twins for Discrete Material Flow Systems: A Review". In *2024 Winter Simulation Conference (WSC)*, 3070-3081 https://doi.org/10.1109/WSC63780.2024.10838729.

Schwede, C. and A. Freiter. 2024. "OpenFactoryTwin (OFacT): Initial Release (0.1.0)". Zenodo https://doi.org/10.5281/zenodo.13734527, accessed 11th April 2025.

Smith, J. K. and C. Dickinson. 2022. "Discrete-event Simulation and Machine Learning for Prototype Composites Manufacture Lead Time Prediction". In *2022 Winter Simulation Conference (WSC)*, 1695-1706 https://doi.org/10.1109/WSC57314.2022.10015322.

Sousa, A., L. Ferreira, R. Ribeiro, J. Xavier, A. Pilastri, and P. Cortez. 2022. "Production Time Prediction for Contract Manufacturing Industries Using Automated Machine Learning". In *18th IFIP WG 12.5 International Conference Proceedings Part II, AIAI 2022*, June 17th-20th, Hersonissos, Greece, 262–273.

Yamashiro, H. and H. Nonaka. 2021. "Estimation of Processing Time Using Machine Learning and Real Factory Data for Optimization of Parallel Machine Scheduling Problem". *Operations Research Perspectives* 8:100196.

AUTHOR BIOGRAPHIES

CHRISTIAN SCHWEDE is Professor for Big Data Analysis at the University of Applied Science and Arts Bielefeld. He is member of the Center for Applied Data Science and Board member of the Institute for Data Science Solutions. Additionally he is a senior scientist at Fraunhofer ISST in Dortmund resposible for AI in Logistics. His reseach area is the application of data- and model-driven methods to optimize logistics and production. He is specialized in automotive logistics and agent-based dezentralized production systems. His email address is christian.schwede@hsbi.de.

ADRIAN FREITER is a researcher at the Fraunhofer ISST in Dortmund in the Dept. Industrial Manufacturing. His research area is the improvement of simulation-based digital twins through AI/ML applications, reducing the manual effort in the model creation process and improving their prediction accuracy, with the aim of making them more interesting for the broad manufacturing domain. His email address is adrian.freiter@isst.fraunhofer.de.