Multivariate time series forecasting for multi-step prediction of throughput bottlenecks using Long Short-Term Memory networks

[1,2] Nikolai West, [1,3] Jochen Deuse
[1]Institute of Production Systems, Technical University Dortmund, Dortmund, Germany
[2] RIF Institute for Research and Transfer e.V., Work and Production Systems, Dortmund, Germany
[3] University of Technology Sydney, Advanced Manufacturing, Sydney, Australia
nikolai.west@tu-dortmund.de, jochen.deuse@ips.tu-dortmund.de

***Abstract*—**Manufacturing companies require an effective approach to predicting bottlenecks in order to proactively initiate improvement actions that prevent throughput losses before they occur. This paper presents a novel method for predicting dynamic bottlenecks in manufacturing systems. The approach is based on multivariate prediction of multi-step bottleneck activity, using buffer levels and past bottleneck behavior. A Long Short-Term Memory network is used to accomplish the prediction. The approach is presented descriptively using a simulated manufacturing system consisting of five stations and six buffers. The discrete event simulation, all generated manufacturing data, as well as the full code of the analysis, are made freely available in the documentation for this paper. The exemplary application of the case study shows that the proposed approach is effective in predicting future bottleneck locations. In a ten-fold cross-validation, the prediction achieves an average ratio of 95.23% correctly predicted bottlenecks for a forecasting horizon of 30 steps. To make the results comparable and generally usable, four comprehensive benchmarks are presented with approaches that do not use Machine Learning methods. To summarize the research, the paper provides a contribution to the manufacturing industry by enabling proactive decision-making for process improvement and optimization.

Keywords—*Active period method, bottleneck prediction, long short-term memory networks, time series forecasting.*

I. INTRODUCTION

Data science tools and methods have a long tradition in predicting various metrics and events in manufacturing, such as the wear-related behavior of technical equipment in maintenance [1]. This area is often referred to as *Predictive Maintenance* and its potential is undisputed due to numerous successful implementations in the manufacturing industry. A closely related approach in terms of the data analytical challenges is the *Bottleneck Prediction*. While this is not yet as widespread as predictive maintenance, it also has significant potential to save resources in the manufacturing industry [2]. In order to achieve long-term production efficiency, companies must ensure flexibility with regard to high product variety, while at the same time handling the increasing complexity of production planning and control.

For state-of-the-art manufacturing companies facing these challenges, Bottleneck Prediction is of particular relevance. According to the *Theory of Constrains*, the performance of any system is inevitably limited by exactly one bottleneck [3]. Only measures to reduce this bottleneck's effect provide an effective way to increase systemic output since any measure taken on any other section in the value stream do not increase the overall throughput.

The dynamic nature of bottlenecks in manufacturing presents additional difficulties in dealing with them. Instead of being confined to only one station, the bottlenecks in manufacturing systems often switch between different locations. This behavior is called *Shifting Bottlenecks* and is caused by variability-related effects in all but the most simplistic real-world systems [4]. Various approaches exist in the literature for a real-time identification of bottlenecks that can account for dynamic behavior, which we briefly discuss in **Chap. II**.

In practice, identifying a bottleneck is not sufficient to achieve any improvements of efficiency. Just as in Predictive Maintenance, the identification constitutes only as a reactive measure. E.g. wear-related equipment failure is accompanied by losses in availability and throughput. The same applies to the occurrence of bottlenecks: The presence of a bottleneck is inevitably associated with a limitation of the entire system due to the fundamental assumption of TOC. Any corrective measures to a bottleneck after its identification is therefore always a reactive one. Maintenance has shown that such predictive measures can have a significant advantage over all reactive responses. To achieve a similar benefit for the field of bottleneck analysis, the objective of this paper is to contribute to the development of a method for the prediction of dynamic bottlenecks.

The remainder of this paper is organized as follows. First, we describe the relevant fundamentals of modern bottleneck analysis (**Chap. II**). In this respect, we address a holistic model that divides the tasks of bottleneck analysis into four steps. In addition, we also present a brief summary of the most important scientific work on bottleneck prediction from recent years. Then, we introduce the design of the simulation, on which further work is based (**Chap. III**). In addition to the layout of the simulated manufacturing system, we present in particular the selected parameterization of the discrete event simulation model and the resulting data later used for the bottleneck prediction. We then explain our approach to predicting dynamic bottlenecks using a machine-learning model (**Chap. IV**). Using a simple example, we outline our considerations for the use of buffer levels and present two analysis scenarios with different input data for model training. We then present the results of the modeling and compare the two analysis scenarios with four custom benchmarks (**Chap. V**). The approach of the second scenario scores best and manages to correctly predict a bottleneck for a forecasting horizon of 30 steps with an average ratio of 95.23%. Finally, we present a brief summary of our work and consider next steps and further research needs in the field of bottleneck analysis (**Chap VI**).

II. FUNDAMENTALS

A. Steps of holistic bottleneck analysis

Bottleneck analysis involves several tasks that have varying objectives and for which different approaches have been proposed. The degree of maturity of the respective research areas of these tasks varies significantly. In the following, we enumerate the four key steps of a holistic bottleneck analysis [5]. The four steps categorize the tasks required in the process in a successive manner and have already been addressed and further refined in promising related research [6].

- **Bottleneck detection**: As the first step, it determines the current location of the bottleneck in the manufacturing system. It can utilize different metrics, such as machine states, buffer levels or process times, and allows for a concise determination of the bottleneck. Approaches are subdivided into momentary value methods and average value methods, according to the respective aggregation.
- **Bottleneck diagnosis**: As the second step, it utilizes the knowledge of bottleneck locations from the first step to assess the frequency and severity of bottleneck behavior to the manufacturing system. In addition to examining the effects, this step also includes a cause analysis of the bottlenecks, for which past system data is combined with experience-based knowledge.
- **Bottleneck prediction**: As the third step, it involves the prediction of future bottleneck locations. A prerequisite is the implementation of an appropriate bottleneck detection that provides the data basis for predicting future system behavior. As a field of research, it can be described as not yet fully explored, which is why it is attracting a considerable degree of attention in the ongoing scientific discourse.
- **Bottleneck prescription:** As the fourths step, it culminates all previous results in an independent and self-improving control system. Based on predicted bottlenecks, it enables not only periodization but also initiation of remedial measurements. Due to the high complexity and relative novelty of this step, no proven approaches to its implementation exist yet and the need for research is to be considered substantial.

To summarize, these four steps enable a holistic bottleneck analysis. Since the steps are arranged in a successive manner, completion of the previous steps is expected before the respective next step can be addressed. However, since the focus in this paper is the third step, the bottleneck prediction, we cover the previous two steps only rudimentarily.

We leverage the *Active Period Method* (APM), already mentioned in the introduction, as the detection method in this paper. APM relies on the assumption that the station with the longest period of uninterrupted activity is the bottleneck. Since all other stations are more often blocked by subsequent buffers filled to the maximum as a result of the bottleneck, or starved by lack of replenishment from previous stations, their active operational periods are correspondingly shorter. For an in-depth investigation for the usage of bottleneck detection methods in serial manufacturing lines, including APM, we refer to [2]. Additionally, for a detailed description of more methods for bottleneck detection we refer to [7–9].

B. Related contributions on bottleneck prediction

Bottleneck prediction helps to anticipate the future location of a bottleneck so that measures can be taken proactively before it causes any drop in throughput. In the following, we provide an overview of how research on bottleneck prediction has unfolded in the past few years. The selection is build up-on [5] and takes into account the review of [10]. For better orientation, **Table 1** summarizes the relevant contributions.

Table 1. *Overview of related work on bottleneck prediction*

	Data	**Method**
[11]	Blockage and starvation times	ARMA
[12]	Utilization, buffer level, cycle times	ANFIS
[13]	Buffer level for shifting behavior	Probability
[14]	Machine states on a shift basis	ARIMA
[15]	Machine states, cycle time, product mix	LSTM
[10]	Utilization	LSTM
[16]	Blockage and starvation times	GAT

LI ET AL. (2011) propose an approach based on time-series prediction that uses an *Autoregressive Moving Average* (ARMA) model to predict the future blockage and starvation states of machines [11]. On a simulation with 17 stations, they succeed with 97.38% accuracy in predicting the bottleneck in the next shift, i.e. in an eight-hour period. CAO ET AL. (2012) use an *Adaptive Network-based Fuzzy Inference System* (ANFIS) to incorporate a variety of simulation data, such as utilization rate, buffer level and cycle times, in the bottleneck prediction [12]. Their artificial neural network uses one-hot encoding and an n-input-1-output setup to predict the station numbers of future bottlenecks by utilizing a softmax layer, performing with about 90% accuracy. ROSER ET AL. (2017) developed an approach that determines the likeliness of a bottleneck shift based on the currently available material in the systems' buffers [13]. The approach extends APM by a wandering observation method, which allows a determination of shifting probabilities of bottlenecks, based on buffer levels.

SUBRAMANIYAN ET AL. (2018) predict the active period of machines, but use an *Auto-Regressive Integrated Moving Average* (ARIMA) method [14]. In an automotive use case with 13 stations, condensed machine states from 315 shifts allow a bottleneck prediction with an accuracy of 89.2%. LAI ET AL (2018) utilize a *Long Short-Term Memory* (LSTM) network with two layers to predict bottlenecks in a complex, dynamic multi-job manufacturing system [15]. In direct comparison with ARMA, LSTM improves the bottleneck prediction that is based on cycle and blockage times. LIN ET AL. (2022) propose another LSTM-based prediction method that relies on an integrated temporal unidirectional and spatial bidirectional structure [10]. With the help of this model, it is possible to accurately predict the utilization in different simulation scenarios with eight to 12 stations. LAI ET AL. (2023) develop an interpretable modeling framework that utilizes spatial and temporal dynamics to predict bottlenecks by leveraging *Graph Attention Networks* (GAT) [16].

Finally, we would also like to highlight the recent work by ROCHA AND LOPES (2022) that provides a comprehensive comparison of eleven prediction methods [17]. Their approach relies on the prediction of process times, with *Random Forest* (RF) and *Multi-Layer Perceptron* (MLP) emerging as the most suitable methods.

III. STUDY DESIGN

A. Layout of the manufacturing value stream

The success of past work is often difficult to compare due to the specificity of the respective use cases. Often, performance metrics are used to evaluate the prediction without providing any reference to the real detection of bottlenecks. In addition, the quality of a correct forecasting depends to a very high degree on the actual dynamics of the considered systems. For this reason, we will demonstrate the approach presented in this paper with the help of a simple and illustrative example. We also provide the entire source code for analysis [18].

In this chapter, we will therefore first present the design of the case study before introducing our approach for bottleneck prediction with LSTM in the next section (**Chap. IV**). Our goal is to create a comprehensible and assessable bottleneck prediction that unmistakably demonstrates the potential and that can serve as a template for practical implementations.

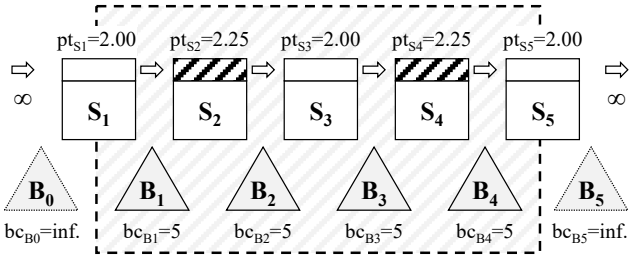

Figure 1. *Visualization of the simulated manufacturing line*

Figure 1 shows the layout chosen for the manufacturing simulation. The flow line consist of five fully interlinked work stations S_i that are interconnected by a total of six buffers B_i. For each S_i, a process time pt_{Si} is specified and each B_i is constrained a maximum capacity bc_{Bi}. The boundaries of the system are unrestricted, which corresponds to an unlimited supply of material for S_1 and unlimited customer demand for S_5.

B. Parameterization of the simulation

To create a scenario in which dynamic bottlenecks occur, adjustments we made for pt_{S2} and pt_{S4}. While the base value of the process times for all stations is 2.00, S_2 and S_4 received an addition of 12.5% so that the respective pt is 2.25. These adjustments enforce a shifting behavior of bottlenecks, especially between S_2 and S_4.

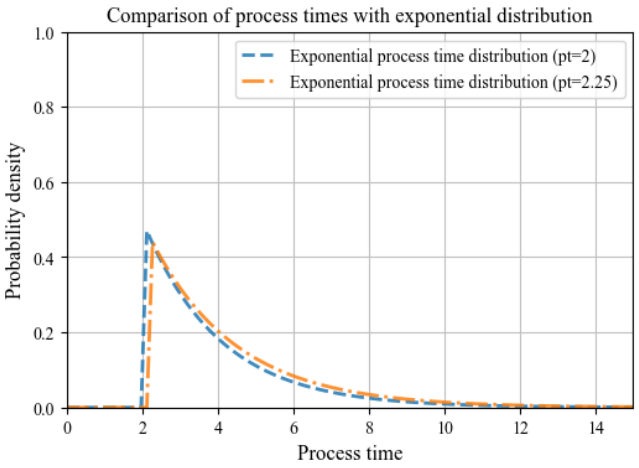

Figure 2. *Process time distribution in the simulation.*

Without the introduction of variability, the systemic behavior remains deterministic and does not correspond to any real manufacturing data. For this reason, we introduced an exponential distribution to the base pt of 2.00 and 2.25, respectively. **Figure 2** illustrates the resulting probability distribution of process times. While a low and consistent pt is often the rule in manufacutring, random events may cause significant delays. This distribution corresponds to the observable behavior and was thus chosen for the simulation.

The simulation was created using the Python library SIMPY [19] and the entire simulation data as well as the created simulation can also be found in the project repository [18].

C. Simulated data for the prediction

Table 2 shows a summary of the data. First, the length of a simulation was set to 10,000 steps. To ensure robustness, we performed 1,000 simulations. The simulations provide the events in the value stream, i.e. primarily the completion of orders at stations, as well as the respective level of each buffer. In accordance to APM, we determined the active periods of the stations and derived the information on the locations of the bottlenecks.

Table 2: *Summary of the simulation data*

Data set	Amount	Lengths
Simulation events `events_10k_S2-S4+25%_XXX.csv`	1.000	10.000
Buffer level `buffer_10k_S2-S4+25%_XXX.csv`	1.000	10.000
Active periods `active_periods_10k_S2-S4+25%_XXX.csv`	1.000	10.000

Figure 3 illustrates an exemplary section of a simulation. The data was arbitrarily selected from the first recording. The upper diagram shows the buffer levels of B_1 to B_4. The values are discrete and range from zero to the selected bc_{Bi} of 5. In this visualization, but also in the further course of the analysis, we refrain from using B_0 and B_5. Due to the assumption about the system boundaries, these buffers have predetermined values and do not provide additional insight into the dynamics in the system. Next, the middle diagram depicts the respective active periods of S_1 to S_5. The curve progressions have a saw blade shape. Values always increase continuously before falling back to zero after an interruption of the active period. Finally, the lower diagram shows the respective bottleneck. Since a bottleneck is determined by the longest active period, any bottleneck shift corresponds to a saw blade tip of the active period from the middle diagram.

The relative percentage of occurrence as bottleneck of each station provides a better understanding of the dynamics of the system. As such, we determined the ratios for all stations as an average of the 1,000 simulations. With regard to the entire simulation, S_2 and S_4 occur with the greatest frequency as bottlenecks with 44.13% and 41.22% of the time, respectively. The other stations appear correspondingly less frequently as bottlenecks, with 5.41% for S_1, with 5.50% for S_3 and with 3.72% for S_5.

These values help to determine a numeric target for the bottleneck prediction. A naive prediction that always expects S_2 to be the bottleneck would achieve an average accuracy of about 44%. Thus, to provide additional value, the prediction in this system must perform better than said naive approach.

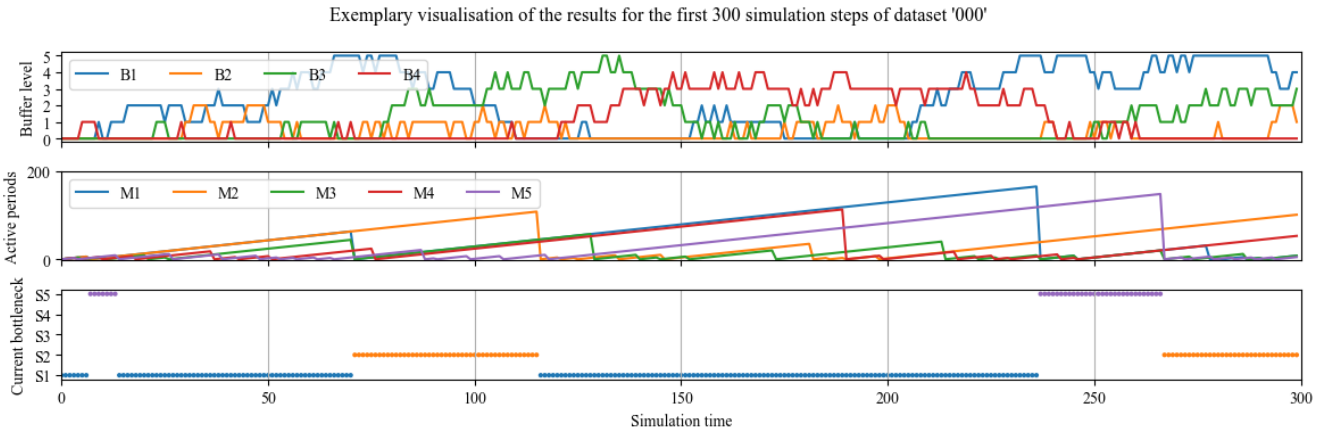

Figure 3. *Exemplary visualization of the simulation results for the first 300 steps of dataset number '000'*

IV. Approach

A. Introduction of LSTM networks for bottleneck prediction

As introduced in **Chap. II**, LSTMs have enabled various accomplishments in bottleneck prediction [10, 15]. In this paper, we refrain from a mathematical description of the functionalities of LSTM networks and restrict ourselves to a practical explanation of the core concepts. For further information, we refer to the referenced works in **Table 1**.

LSTM are a type of *Recurrent Neural Network* (RNN) that are specifically designed to overcome the vanishing gradient problem that occurs with traditional RNNs [20]. The key difference between LSTM and traditional RNN is that the former utilizes a memory block to store information, which comprises one or more memory cells and three adaptive, multiplicative gating units shared by all cells in the. The three gates are the input gate, forget gate, and output gate, which are responsible for controlling the flow of information into, out of, and within the memory block [21]. The input gate of an LSTM network is trained to learn what information should be stored in the memory from the historical data, while the forget gate is used to learn how long the information should be stored. The output gate is trained to learn when to read the memory out. This makes LSTM networks particularly useful for time-series forecasting, as they can selectively remember or forget information from the past based on its relevance to the current prediction [22]. As a result, LSTM networks can effectively model the long-term dependencies that are often present in time series, making them ideal for bottleneck prediction [20]. However, one disadvantage of LSTMs is that they can be computationally expensive to train, especially when dealing with large datasets or complex models. Therefore, in the provided documentation [18], there are already trained models that can be used to reproduce the prediction more quickly .

B. Approach for bottleneck prediction using buffer levels

Our declared goal is to predict future bottleneck behavior using an LSTM network. Since we have chosen the APM as most suitable bottleneck detection method, the investigation of the respective buffer levels is the logical choice. That said, the following approach is based on the simple hypothesis that the trends and developments of buffer levels allow us to gain insight into the future of a bottleneck's location:

- For one, we expect that a *continuous decline* in material of the buffer in front of the bottleneck will indicate an upcoming *starvation* of the respective station, which in turn is accompanied by a bottleneck shift.
- We also expect that *approaching the maximum capacity* of a buffer will signal an imminent *blockage* of the respective upstream station. If said station is the current bottleneck, a bottleneck shift is expected to happen soon.

The hypothesis led to the developed LSTM for bottleneck prediction. To provide a better illustration of the models input structure, **Figure 4** displays exemplary data from the study, which we will use to further outline our approach.

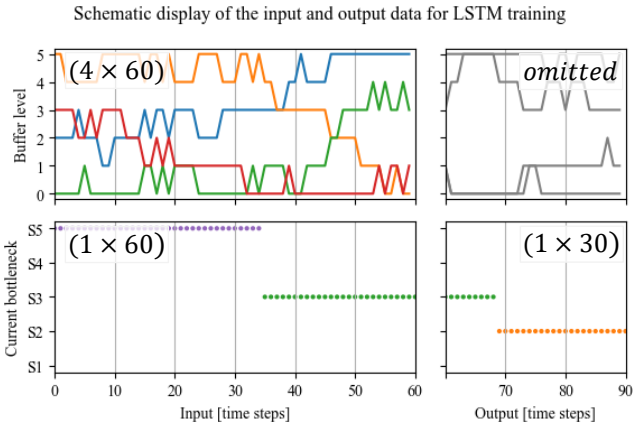

Figure 4. *Schematic display of the available input and output data from the simulations for bottleneck prediction*

First, a period has to be defined as input for the prediction. In the example in **Figure 4**, as well as in the later analysis in **Chap. V**, an interval of 60 simulations steps was chosen, which may correspond to the next hour in a real application, for example. Additionally, a forecasting horizon has to be chosen as well. As shown, we have chosen 30 steps for this, which could correspond, for example, to a forecast of half an hour. We have based these values on the distribution of process times (see **Figure 2**). Since the level progression of multiple buffers is available, the LSTM network has to take these multivariate inputs (4×60). Knowledge about future buffer levels has no direct value for the bottleneck prediction $(omitted)$. Instead, we utilize the station numbers as categorical output values (1×30). In order for the

categorical values to be used in the LSTM, we use one-hot encoding to encode the stations to be predicted. Since there are five stations, the predicted bottleneck has five possible outcomes and the output layer is a one-dimensional vector of length 150. In addition, information on past bottlenecks is also available at the time of prediction (1×60). This information has a very high degree of potential knowledge, since future bottlenecks are directly dependent on past bottlenecks. To determine the impact of this addition, we consider two scenarios in the following chapter.

- **Scenario 1:** Prediction without past bottleneck states in the training data ($input\ shape = [4 \times 60]$)
- **Scenario 2:** Prediction including past bottleneck states in the training data ($input\ shape = [5 \times 60]$)

We expect the forecast of the Scenario 2 [5×60] to be better than of Scenario 1 [4×60] due to the additional information.

C. *Further assumptions for model parameterization*

The same modeling parameterization was chosen for both scenarios. The Python-based library KERAS, which is built on the machine learning platform TENSORFLOW, was used to create the model (in Version 2.4.0). With two hidden layers that consist of densely connected 64 neurons, the model can still be described as comparatively simple. The activation functions were set to *tangent hyperbolic*, and *Adam* was chosen as optimizer with a typical learning rate of 0.0001. To focus on the bottleneck prediction, we refrain from an in-depth description of the model, but instead refer to the documentation [18] for more details (see *prediction.py*).

V. RESULTS AND DISCUSSION

A. *Results of the bottleneck prediction*

This chapter now presents the results of the two analysis scenarios. To do this, we will first look at the course of the training, with **Figure 5** showing an exemplary training run.

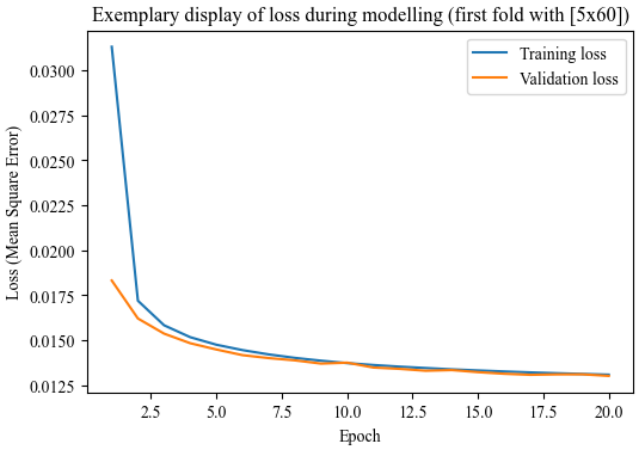

Figure 5. *Exemplary visualization of the loss values during training of the first fold with the [5x60] input data*

The training was conducted for a total of twenty epochs and the *Mean Squared Error* (MSE) was used as the loss measure. In the visualization, the effects of the training can be seen well. After the twenty epochs, there was either no more improvement or the validation loss started to decrease due to overfitting. Based on the low MSE, we summarize that the model achieves a stable result comparatively fast.

The ability to recognize the current bottleneck is crucial to prove the success of the approach. Since the results may vary due to the stochastic nature of the modelling, we perform a *ten-fold cross validation*. In this validation method, a data set is divided into ten random parts. Nine parts are used to train the model, while the tenth part is used for testing or validation. In this way, ten results are obtained, mitigating stochastic deviations [23]. **Figure 6** displays the results of this approach for both scenarios, with and without the bottleneck information in the training data (Scenario 1 and 2).

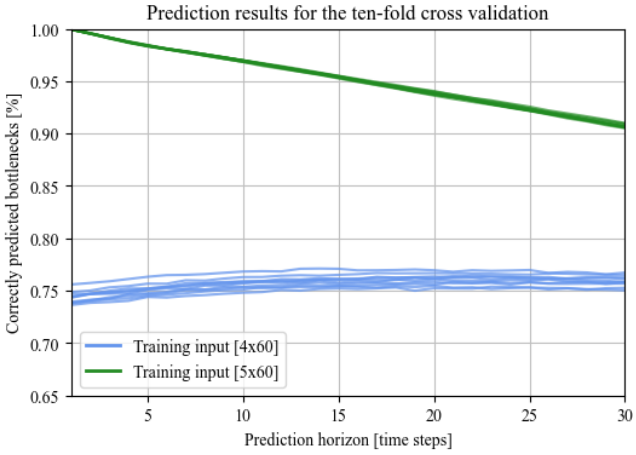

Figure 6. *Results of the bottleneck prediction of all ten folds for each of the two selected scenarios*

First, consider the set of curves for the ten predictions of Scenario 1 (4×60). For each of the thirty steps in the forecast horizon, the model has designated exactly one station as a bottleneck. **Figure 6** shows the percentage of bottlenecks correctly predicted at each step of the prediction horizon. For all ten folds, the percentage value is comparatively stable at about 75%. There are no significant deviations or outliers and all prediction results are within a few percentage points of each other. Interestingly, the trajectories show a slight trend toward upward curvature. At the beginning of the forecast horizon, the worst prediction accuracy is achieved at about 74%. In contrast, slightly better accuracies can be achieved in the middle and end of the prediction horizon with about 77%. To summarize, the approach of Scenario 1 succeeds in predicting the bottleneck in about three-quarters of all cases.

As postulated in **Chap. IV**, predictions from Scenario 2 (5×60) achieve even better results. Again, a tenfold cross-validation was performed, but in this case, the outcomes were much closer. Starting from nearly 100% correctly predicted bottlenecks in the first point of the prediction horizon, the percentage decreases in an approximately linear fashion. The prediction then reaches the lowest percentage in the last step of the prediction horizon of 30 steps and at about 91% correctly predicted bottlenecks. Such a progression seems natural and was to be expected: With each step into the future of the prediction horizon, the difficulty to make a correct prediction increases, which is made more difficult by unforeseeable events. In our study, an unexpectedly high process time due to the exponential distribution would constitute such an irregular event. Due to the high percentage of correct predictions, the application of Scenario 2 can be considered successful.

Now, it was said at the beginning of **Chap. III** that any evaluation of a bottleneck prediction depends to a large degree on the specific circumstances of the particular use

case. For this reason, we conduct an in-depth examination of the results based on several benchmarks in the next section.

B. Discussion of results with benchmarking

As benchmarking, we now outline four approaches to ensure a better assessment of the results of the bottleneck prediction.

- **Random:** As there are five stations in the study, there are five possible specifications for the bottleneck prediction. As the simplest approach, the random benchmark selects a station number as the bottleneck for each point of the prediction horizon without any further considerations.
- **Only S2:** By the study design, it was determined that bottlenecks were more common at station $S2$. With 12.5% additional process time, the station limits the entire system more frequently and has a correspondingly higher probability of occurring as bottleneck. We addressed this behavior at the end of **Chap. III** and we now incorporate it in a benchmarking approach that only predicts $S2$ for the entire prediction horizon.
- **Only S4:** The approach is analogous to '*Only S2'*, but here S4, the second station with additional process time, is predicted as bottleneck during the entire horizon.
- **Last bottleneck:** The aforementioned benchmarking approaches are comparatively naive, since they do not use implicit knowledge about the system behavior. In this approach, the last bottleneck in each training data set is used as a prediction for the next steps. Since shifting bottlenecks do occur, but are comparatively infrequent, this approach has a reasonably high chance to correctly forecast the prediction horizon.

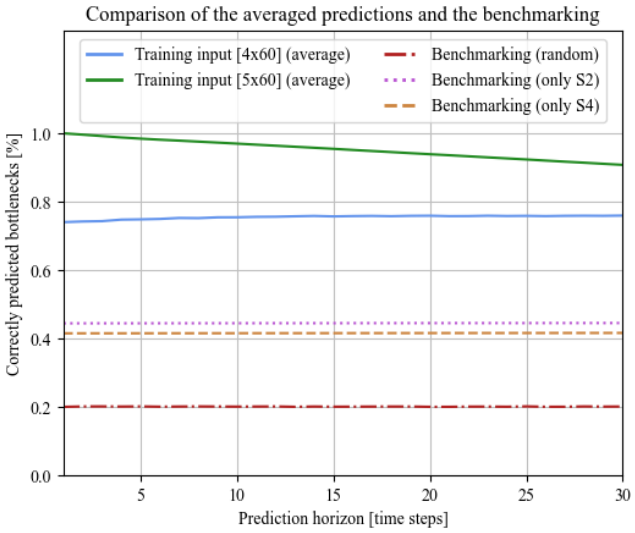

Figure 7. *Comparison of the prediction results for Scenario 1 and 2 and three benchmarking approaches*

Figure 7 again shows the prediction result of the Scenarios 1 and 2, whereby for the sake of clarity only the averaged result of the ten cross-validations is displayed. In addition, the figure shows the predictions of the three benchmarks '*random*', '*only S2*' and '*only S4*'. The Scenario 1 and 2 predictions are unchanged, again showing a nearly constant percentage in Scenario 1 of about 75% and a roughly linear decline in Scenario 2. Of the three benchmarks displayed, 'random' shows the worst performance. In accordance with probabilistic chance, the proportion of values correctly predicted as bottlenecks is only 20%. The percentage of correctly predicted bottlenecks with the benchmarks '*only S2*' and '*only S4*' is about twice as high. Again, the accuracy is constant over the entire prediction horizon. It corresponds well to the bottleneck frequencies of 41% and 44% for S2 and for S4 that were anticipated.

Based on an examination of these three benchmarks, it can be said that the predictions made represent a good forecast. In the visualization, however, we have not yet shown the fourth benchmarking '*last bottleneck*' because it is very close to Scenario 2. **Figure 8** therefore shows an enlarged section comparing the results of '*last bottleneck*' to Scenario 2.

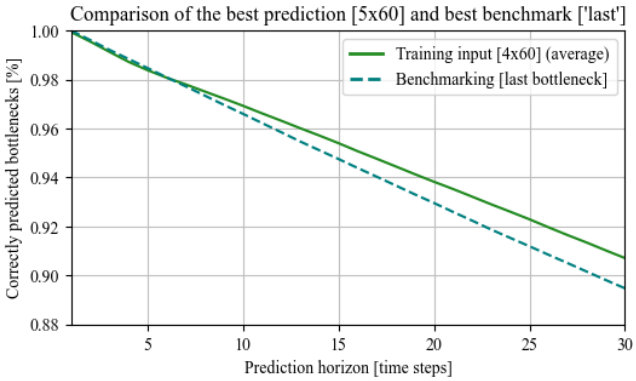

Figure 8. *Comparison of Scenario 2 and the benchmarking approach 'last bottleneck'*

The comparison in **Figure 8** shows that the benchmarking achieves results comparable to those of Scenario 2. It also shows a linear slope that begins with an almost perfect prediction for the first point of the prediction horizon. As for Scenario 2, the ratio of correctly predicted bottlenecks declines with an increasing prediction horizon, dropping to around 89%. Thus, the accuracy of Scenario 2's prediction is better than the results of the benchmarking, starting at a horizon of 5 steps. Towards the end of the prediction horizon, the results of Scenario 2 are about 2% better than the benchmark 'last bottleneck'. The results of scenario 2 can thus still be considered good, but the comparison with this still relatively simple benchmarking has shown that a mere consideration of the proportion of correctly predicted bottleneck states can be a deceptive measure.

Table 3: *Summary of the respective prediction results (averaged for the entire prediction horizon)*

Approach	Correct predictions [%]
Scenario 1	0.7541
Scenario 2	0.9523
Benchmark 'random'	0.1999
Benchmark 'only S2'	0.4440
Benchmark 'only S4'	0.4188
Benchmark 'last bottleneck'	0.9463

Table 3 concludes by summarizing all predictions. The table shows the percentage of average predicted bottlenecks in each case. As discussed, the three benchmarks 'random', 'only S2' and 'only S4' perform the worst. With an average ratio of correct predictions of 75.41%, Scenario 1 reaches an acceptable result. With 94.64% and 95.23% respectively, the benchmark 'last bottleneck' and Scenario 2 achieve by far the best results, with Scenario 2 performing marginally better.

VI. CONCLUSION

In summary, the presented approach enables the prediction of dynamic bottlenecks and fulfills the initially formulated objective. While LSTMs have been used for bottleneck forecasting in the past, the innovation of this contribution lies in the method used to predict future bottlenecks. Instead of predicting manufacturing metrics of the production system, the approach achieves a straightforward forecast of the bottlenecks through a corresponding output layer. The direct comparison of the predictions from Scenario 2 to the results of the benchmark '*last bottleneck*' showcase this benefit.

However, the measure used to determine prediction quality can be improved in future work. In this paper, the percentage of correctly predicted bottlenecks was used as a measure, which typically indicates an accuracy of the prediction. However, the high value achieved by the benchmark '*last bottleneck*' shows that application cases should be treated more like analysis tasks with unevenly distributed classes. For example, **Figure 8** has shown that '*last bottleneck*' can achieve an average prediction accuracy of about 89% after 30 steps into the future. At the same time, this implies that only in about 11% of all training data has one (or more) bottleneck shift occurred. This emphasizes the advantage of Scenario 2, which only improved the prediction rate by a few percent.

These observations, particularly when a bottleneck shift takes place, are of the utmost importance for production planning and control. In further works on bottleneck prediction, we plan to incorporate a suitable 'class weight' that accounts for the fact that bottlenecks behave rather statically in certain intervals.

Lastly, the simplified example of the case study should be mentioned. In the simulation, there is a directed material flow of a completely interlinked flow line. In addition, only one product variant was manufactured. The application serves as a proof-of-concept and must be transferred to more complex manufacturing systems. If this is successful, bottleneck prediction may achieve a similar level of popularity as Predictive Maintenance that we discussed at the beginning.

ACKNOWLEDGEMENTS

This paper is part of the project 'Prediction of dynamic bottlenecks in directed material flow systems using machine learning methods' (PrEPFlow, 21595), which is funded by the German Federal Ministry of Economics and Technology (BMWi), through the Working Group of Industrial Research Associations (AIF). It is carried out on behalf of the German Logistics Association e.V. (BVL) and it is part of the program for promotion of joint industrial research and development (IGF) based on a resolution of the German Bundestag.

REFERENCES

[1] R. Wöstmann, A. Barthelmey, N. West, and J. Deuse, "A retrofit approach for predictive maintenance," in *Tagungsband des 4. Kongresses Montage Handhabung Industrieroboter*, T. Schüppstuhl, K. Tracht, and J. Roßmann, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2019, pp. 94–106.

[2] N. West, J. Schwenken, and J. Deuse, "Comparative study of methods for the real-time detection of dynamic bottlenecks in serial production lines," in *Lecture Notes in Computer Science, Advances and trends in artificial intelligence: Theory and practices in artificial intelligence*, H. Fujita, P. Fournier-Viger, M. Ali, and Y. Wang, Eds., Cham: Springer International Publishing, 2022, pp. 3–14.

[3] Eliyahu M. Goldratt, *The Goal: Excellence in manufacturing*. Croton-on-Hudson, NY: North River Press, 1984.

[4] C. Roser, M. Nakano, and M. Tanaka, "Shifting bottleneck detection," *Proceedings of the 2002 Winter Simulation Conference*, no. 1, pp. 1079–1086, 2002.

[5] N. West, M. Syberg, and J. Deuse, "A holistic methodology for successive bottleneck analysis in dynamic value streams of manufacturing companies," in *Towards sustainable customization: Bridging smart products and manufacturing systems*, 2022, pp. 612–619.

[6] E. Mahmoodi, M. Fathi, and M. Ghobakhloo, "The impact of Industry 4.0 on bottleneck analysis in production and manufacturing: Current trends and future perspectives," *Computers & Industrial Engineering*, vol. 174, p. 108801, 2022, doi: 10.1016/j.cie.2022.108801.

[7] Y. Wang, Q. Zhao, and D. Zheng, "Bottlenecks in production networks: An overview," *J. Syst. Sci. Syst. Eng.*, vol. 14, no. 3, pp. 347–363, 2005, doi: 10.1007/s11518-006-0198-3.

[8] C. E. Betterton and S. J. Silver, "Detecting bottlenecks in serial production lines: A focus on interdeparture time variance," *International Journal of Production Research*, vol. 50, no. 15, pp. 4158–4174, 2012, doi: 10.1080/00207543.2011.596847.

[9] C. Yu and A. Matta, "A statistical framework of data-driven bottleneck identification in manufacturing systems," *International Journal of Production Research*, vol. 54, no. 21, pp. 6317–6332, 2016, doi: 10.1080/00207543.2015.1126681.

[10] L. Ma, T. Qu, M. Thürer, Z. Wang, M. Yuan, and L. Liu, "An integrated spatial-temporal neural network for proactive throughput bottleneck prediction in high-variety shops with complex job routings," *International Journal of Production Research*, pp. 1–13, 2022, doi: 10.1080/00207543.2022.2148769.

[11] L. Li, Q. Chang, G. Xiao, and S. Ambani, "Throughput bottleneck prediction of manufacturing systems using time series analysis," *Journal of Manufacturing Science and Engineering*, vol. 133, no. 2, 2011, doi: 10.1115/1.4003786.

[12] Z. Cao, J. Deng, M. Liu, and Y. Wang, "Bottleneck prediction method based on improved adaptive network-based fuzzy inference system (ANFIS) in semiconductor manufacturing system," *Chinese Journal of Chemical Engineering*, vol. 20, no. 6, pp. 1081–1088, 2012, doi: 10.1016/S1004-9541(12)60590-4.

[13] C. Roser *et al.,* "Bottleneck prediction using the active period method in combination with buffer inventories," in *IFIP Advances in Information and Communication Technology, Advances in production management systems: The path to intelligent, collaborative and sustainable manufacturing*, H. Lödding, R. Riedel, K.-D. Thoben, G. von Cieminski, and D. Kiritsis, Eds., Cham: Springer International Publishing, 2017, pp. 374–381.

[14] M. Subramaniyan, A. Skoogh, H. Salomonsson, P. Bangalore, and J. Bokrantz, "A data-driven algorithm to predict throughput bottlenecks in a production system based on active periods of the machines," *Computers & Industrial Engineering*, vol. 125, pp. 533–544, 2018, doi: 10.1016/j.cie.2018.04.024.

[15] X. Lai, H. Shui, and J. Ni, "A two-layer long short-term memory network for bottleneck prediction in multi-job manufacturing systems," *Proceedings of the International Manufacturing Science and Engineering Conference*, vol. 12, no. 1, 2018, doi: 10.1115/MSEC2018-6678.

[16] X. Lai, T. Qiu, H. Shui, D. Ding, and J. Ni, "Predicting future production system bottlenecks with a graph neural network approach," *Journal of Manufacturing Systems*, vol. 67, no. 1, pp. 201–212, 2023, doi: 10.1016/j.jmsy.2023.01.010.

[17] E. M. Rocha and M. J. Lopes, "Bottleneck prediction and data-driven discrete-event simulation for a balanced manufacturing line," *Procedia Computer Science*, vol. 200, pp. 1145–1154, 2022, doi: 10.1016/j.procs.2022.01.314.

[18] N. West, *Project repository*, 2023. [Online]. Available: github.com/nikolaiwest/2023-bottleneck-prediction-icrcet

[19] Team SimPy, *SimPy Documentation: Release 4.0.2.dev1+g2973dbe*, 2020.

[20] S. Hochreiter and J. Schmidthuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[21] F. A. Gers, N. N. Schraudolph, and J. Schmidthuber, "Learning precise timing with LSTM recurrent networks," *Journal of Machine Learning Research*, vol. 3, no. 1, pp. 115–143, 2002, doi: 10.1162/153244303768966139.

[22] Y. Hua, Z. Zhao, R. Li, X. Chen, Z. Liu, and H. Zhang, "Deep learning with long short-term memory for time series prediction," *Neural and Evolutionary Computing*, no. 1, pp. 1–9, 2018, doi: 10.48550/arXiv.1810.10161.

[23] V. Cerqueira, L. Torgo, and I. Mozetic, "Evaluating time series forecasting models: An empirical study on performance estimation methods," *arXiv:1905.11744v1*, no. 1, 2019.