Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 78 (2018) 190–195

www.elsevier.com/locate/procedia

6th CIRP Global Web Conference "Envisaging the future manufacturing, design, technologies and systems in innovation era"

Online lead time prediction supporting situation-aware production control

Dávid Gyulai[a,*], András Pfeiffer[a], Júlia Bergmann[a], Viola Gallina[b]

[a]*Centre of Excellence in Production Informatics and Control (EPIC), Institute for Computer Science and Control (SZTAKI), Hungarian Academy of Sciences (MTA), Kende 13-17, H-1111 Budapest, Hungary*

[b]*Fraunhofer Austria Research GmbH, Theresianumgasse 27, A-1040 Wien, Austria*

Abstract

Cyber-Physical Production Systems (CPPS) appeared already in recent manufacturing environments, and they are capable of providing detailed data about the products, processes and resources in near real-time. Various analytics techniques are available to exploit such technology related data in decision making, however, these tools typically act in the fields of maintenance and quality. Only a few approaches target production control, while the effectiveness of related processes is of crucial importance from overall performance's viewpoint. In the paper, a new production data analytics tool is presented, applying machine learning techniques to proactively predict manufacturing lead times to make decisions by implementing a closed-loop production control. The proposed method applies regression techniques, and based on that, it supports job priorization to be utilized in dispatching decisions. The efficiency of the proposed method is analyzed and presented by numerical results of a case study.

Keywords: production control; machine learning; data analytics

1. Introduction and motivation

1.1. Machine learning applications in production control

In the past few years, artificial intelligence (AI) has become again an emerging field, not only in the business and IT sectors, but also in the industry. In the era of cyber-physical environments, production systems are developed so as to generate and utilize data, in a way as never before, enabling to make production systems smarter. This becomes possible, in case AI techniques are used for making or supporting complex decisions and performing tasks in an automated way that would otherwise require human interaction or domain expertise. Such means of AI are applied to complete or replace human logic in production management, utilizing the data that is generated by the system in (near) real-time. Although one can obtain detailed data about products, processes and system elements, neither it can be applied directly for decision making without preprocessing and analyzing that, nor it is informative for human decision makers, due to its volume and complexity. Therefore, in order to obtain useful information for decision support, data analytics tools are required, furthermore, automated decision making and prediction ask for machine learning (ML) techniques.

In general ML is a subset of AI, and a common applied way of achieving that. It provides the capability for processing large and complex data, to build models upon, supporting decisions with forecasts and predictions. Although in production management ML techniques are applied in the domains of quality control and maintenance, there are still only a few successful applications in production planning and control (PPC). However, there are several opportunities for replacing conventional, typically rule-based decision making mechanisms with data-driven ones, providing the opportunity for reacting on various production situations in a faster and more efficient way.

In the paper, a new, ML-based approach is presented that aims at implementing situation-aware production control by continuously collecting and analyzing real-time streamed production data, and supporting decisions with prediction of control parameters by applying periodically trained learning models. The method is presented through a realistic case study, in

* Corresponding author. Tel.: +36-1-279-6181; fax: +36-1-466-7503.
E-mail address: david.gyulai@sztaki.mta.hu (Dávid Gyulai).

10.1016/j.procir.2018.09.071

which manufacturing lead times (LT) are predicted in a flow-shop environment, and complexity of the prediction is resulted by the diversity of process parameters, and by the high influence of customer order stream on the jobs in progress. The individual job LTs are predicted by ML techniques, based on the data of products and processes, obtained from the manufacturing execution (MES) system. Special emphasis is put on the online prediction, which means that lead times are predicted upon the arrival of jobs in the system, and in case interaction is needed, e.g., late completion is predicted, control decisions are made in order to keep the target level of key production indicators.

1.2. Towards situation-aware control: the role of ML in PPC

The method presented in the paper is the continuation of previous works, and each of those are parts of an ongoing research in the topic of situation-aware production control. About the latter, as the authors believe, it will be core element of the future PPC approaches. In this direction, the authors presented first regression techniques to predict manufacturing lead times in a flow shop environment, applying linear and tree-based models [15, 10]. The results reinforced that ML approaches can outperform analytical ones in LT prediction. The implementation of the *digital data twin* was the continuation of this work, highlighting the importance of model retraining frequency that significantly influences the prediction accuracy [7]. These results will be also utilized throughout the present paper, and form the basis of the presented approach to complete the previous works by implementing a closed-loop control, feeding back the results to support the decision making process, so as shifting from offline predictions to online ones.

Latest results of data science and machine learning also mark this direction, providing real-time, distributed (and open-source) data processing frameworks such as Apache Flink® [14] and Apache Storm™[2]. Together with the Internet of Things (IoT) frameworks—either considering commercial or open-source ones—, these software tools form the basis for implementing next generation PPC methods that can react on the changes and disturbances (happen to the manufacturing environment) in near real-time. Such solutions provide the opportunity of identifying various situations, moreover, they are also capable of making decisions to manage, avoid or achieve them, thus implementing prescriptive data analytics, a subsequent step of predictive analytics [3].

Current PPC approaches typically rely on rule-based decisions, without exploiting useful information from the available production data. Although promising approaches have recently been proposed for data-driven control [12, 19] and predictive scheduling [5], still there are open questions to replace traditional, rule-based systems with adaptive ones. In the above characterized situation-aware production control, decisions must rely on the combination of static data (e.g., product attributes) and event-driven data. The latter changes dynamically together with the production situations, therefore, conventional data processing methods are not always capable of dealing with it, but special tools and methods are needed. Event-stream processing (ESP) and complex-event processing (CEP) technologies are designed for tackling such challenges [6], therefore, they will be also important elements of advanced, new generation PPC applications.

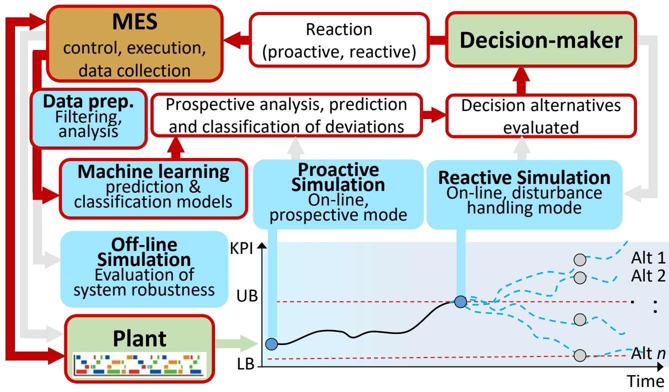

Fig. 1. Combination of simulation and machine learning: a conceptual figure about a generic framework for next generation PPC systems by [15], highlighting the workflow considered in the present paper.

In the paper, a simple production control problem is investigated, namely to reduce lateness of job completions by adjusting the job priorities, based on ML-based prediction of manufacturing LTs. Although this example might seem overly simplistic, the emphasize in the paper is put on the implementation of a closed-loop production control (as first proposed in [15]) by the utilization of future-related ML predictions in control decisions, rather than on the decisions themselves. As illustrated in Fig. 1, the decision maker utilizes MES data (gathered from the plant), applying machine learning to predict the outcome of various decisions. The implementation of this method is a preceding step of a complete situation-aware decision making mechanism, and marks the future direction of the related research towards more sophisticated applications.

1.3. Prediction of manufacturing lead times: state-of-the-art

The essence of the proposed approach relies in the accurate prediction of manufacturing LTs, based on production data gathered in near real-time. Traditionally, expected LTs of jobs are calculated by applying Little's law, considered as the most fundamental analytical method in production control. It states that the average number of items L in a queuing system equals the average arrival rate of items to the system, λ, multiplied by the average waiting time (or lead time) W of an item in the system, thus $L = \lambda W$ [11].

Although Little's law exists for decades now and proven to be very efficient, in recently applied manufacturing systems with dynamically changing parameters and great variety of products produced, lead times are affected by several parameters that increase the complexity of prediction [7]. In order to tackle these challenges, data analytics and ML methods are often applied for lead time prediction. Related promising approaches apply various prediction techniques, e.g., regression trees [13], support-vector machines [1], deep neural networks [21] or linear regression models [17]. These methods are proven to be efficient, however, still there is no rule of thumb how one should select a tool to predict LTs, therefore, domain expertise is of very high importance to achieve success.

2. Problem statement

2.1. Production environment

In the paper, a flow-shop manufacturing environment is analyzed, however, in order to demonstrate the proposed situation aware-control method, the realistic simulation model of the system is applied as a testbed. This model represents efficiently the processes, as in addition to simulating the resources and material flow, it includes stochastic parameters and random events that are also characteristics of real systems. Therefore, throughout the following sections, experiments are performed by using the simulation model, however, the emphasis is not put on the application case, but rather on the ML-based control method and framework that is designed to be generic. Accordingly, it could be apply with any real manufacturing system where similar problems arise, and streamed MES data —applied in the presented method— is available.

In the analyzed flow-shop environment, jobs are processed individually, without applying batches as it often happens e.g., in customized mass production. All products pass through all stages of the system, however, alternative resources are available at each stage to perform a given operation. Operations are performed by machines, and human operators support the processes (e.g. replace parts in the machines), therefore, processing times have a certain deviation. The system consists of three processing stages with 4-3-3 alternative resources at each stage, respectively. The resources have dedicated buffers, in which jobs are waiting to be processed. From production control viewpoint, important to highlight that these buffers are not FIFO-queues but sorters, where job sequences can be altered in case a control actions demands for that. A typical example is preemption: in case a job with higher priority arrives in the buffer, it can be placed in the front of the queue to be processed next, so as reducing its lead time to be completed before the due date.

Five different product types are produced in the system, which differ in the routings and processing times at the different stages. As for the routings, each job is assigned to a machine at all stages applying probability distributions, therefore, each product type can be processed by all machines at a given stage, while they have a characteristic routing that they follow with the highest probability. The processing times are also stochastic, further increasing the complexity of accurate lead time prediction. Additionally, a functional test is performed at the last stage, in order to identify fail products. Following the realistic case, products are randomly marked as fail items, following a uniform distribution, and the average failure rate is 5%. The fail items are not scrapped, but they are reworked at a dedicated station, decoupled from the line. The reworked items are then retested at the last stage, therefore, only functionally correct parts can left the line.

2.2. Lead time prediction problem

Prediction of manufacturing lead times is in the scope of the paper, as in general, LT is one of the most important parameters in production control: it influences various decisions, including due date assignment of jobs, selection of routings, and adjustment of job priorities. In an ideal case, the lead time would be known a-priori when releasing a job in the system, and based on this information, typical problems like unbalanced resources, high work-in-progress (WIP) and late job completion could be avoided. However, in practice, lead time is influenced by several dynamically changing factors, often making it complicated to be predicted accurately. Hence, the emphasis is put on the prediction of job lead times as control-driver parameters, at the time when a job is on entering the system (release).

This can be performed by having information about historical job completions including job features and their lead times, and also knowing the actual state of the system—that always changes dynamically—when a given job is to be released. In the paper, the LT prediction is applied by collecting data about the jobs' and the system's state, synthesizing them into a single dataset. This training set includes static data related features (coming from the ERP), e.g., the jobs' parameters, and also dynamic, event-based ones provided by the MES. As for the static data, a job k is defined by its product type p_k, that is a factor with five levels (five product types are produced). Additionally, from LT prediction perspective, important job parameter is the priority n_k that is an integer number in the range 1-10, and the higher number denotes the larger priority. As mentioned earlier, priority is the basis of sorting jobs in the buffers, influencing the lead times in this way. Important to note that n_k is not a static number (although it has a default value), but a result of a decision, therefore, lead times are actively controlled based on the production situations and job features. Important related parameter is the due-date d_k of the job, which denotes the time until job k need to be completed.

As for the other part of the training data, MES provides event-based logs in near real-time. These events are either correspond to jobs or resources. The logs include the start and end times of job processing, including job ID, resource ID, and a timestamp. The events are streamed by the system in JSON [8] format, and processed online by the ML engine. In this way, jobs are always tracked in the system, which information is utilized to predict LTs. Similarly, resource events are also utilized, as it is important in a control decision making process, whether a certain resource is available or not. Resource log events correspond to the machine downtimes, and such an event determines whether a machine breakdown is happened (failure start event), or a corrective action is completed (failure end event).

Based on the above data and information, the problem of lead time prediction—as considered in the paper—is specified as it follows. Given a set of job features as a tuple $F_k = \langle p_k, n_k, d_k \rangle$, and a list $\langle E \rangle$ of events e. The task is to predict the lead time t_k^L of a job k before it enters the system, utilizing the information obtained from $\langle E \rangle$, and knowing F_k. The lead time t_k^L is defined as the timespan a job spends in the system, between its arrival t_k^a and completion t_c^k. As for the related control problem, the task is to minimize the overall lateness Λ of jobs by adjusting their priorities, in the know of the predicted lead time and the due date (target completion time). In case a job is predicted to be completed later after its due date, its priority is adjusted, so as to make it proceed faster in the system. This adjustment is done by real-time predictions, continuously processing the MES log data, and monitoring the jobs' status information. The overall lateness as a target metric is defined as follows, for a set K of job k: $\Lambda = \sum_{k \in K} t_c^k - d_k = \sum_{k \in K} (t_a^k + t_k^L) - d_k$. Therefore, the key of minimizing the total lateness is predicting accurately the values of t_k^L.

2.3. Descriptive statistical analysis

As part of the presented work, a descriptive statistical analysis was performed first in order to explore the data, so as gaining insights about the operation and behavior of the system. The orders arrive in the system with a random inter-arrival time, resulting in diverse LTs (when no priorization is applied). Observing the histogram of LTs in Fig. 2, one can infer that t_k^L cannot be predicted simply by taking mean values of LTs, as the deviation of those is rather high and their distributions do not follow any statistical pattern.

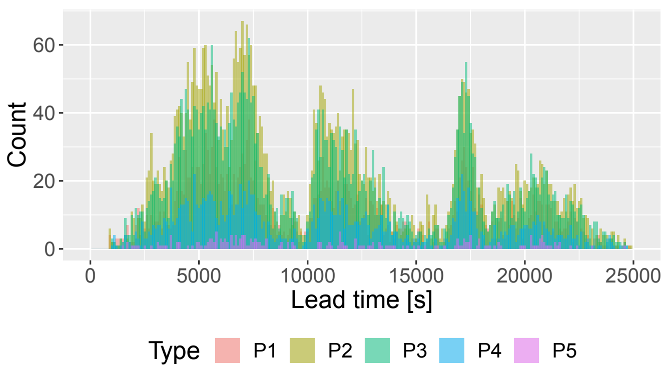

Fig. 2. Histogram of lead times, colored by product type.

Although this fact makes the prediction more complicated, Little's law can still be applied in its generic form, in case the order arrival process is stationary. Considering the order arrival rate as a series of time, stationarity of this process defines that it remains around a statistical equilibrium with probabilistic properties that do not change over time, in particular varying around a constant mean level and with constant variance [4]. Therefore, stationarity test was performed to analyze the arrival process of the jobs: in case the process is stationary, dynamic LT prediction methods —especially those rely on Little's law— can be applied with higher probability of success. For the stationarity analysis, the *ADF* [18] and *KPSS* [9] tests were applied using *tseries R* package [20]. According to the test results, order arrival process is stationary around a mean over the entire horizon, therefore, Little's law can be applied in its canonical form for LT prediction. In the experimental section of the paper, this analytical prediction will provide the baseline of comparison with the ML-based approach.

3. Closed-loop control with online LT prediction

This section provides the overview of the proposed, closed-loop control method with real time prediction of LTs. Before describing the proposed architecture and workflow, the ML models are introduced, utilizing the results of offline experiments first presented by the authors in [7].

3.1. Utilization of offline machine learning results

In the aforementioned paper, offline experiments were performed to investigate the accuracy of lead time prediction, applying different machine learning models. The test results show that both linear and tree-based models could provide reasonable accuracy. The preference in selecting the model to be deployed in an online prediction method relies on two main aspects: on the one hand, linear models provide reasonable accuracy, and their greatest advantages are the short training time and the ease of interpretation. On the other hand, tree-based models—especially ensemble ones like random forests—provide high prediction accuracy and feature importance ranking, while they are hard to interpret and the training process can be time consuming. Another important conclusion of the previous study was the impact of model retraining frequency on the prediction accuracy: the ML model—implementing the digital data twin of the system—need to be retrained periodically, in order to represent the actual state of the physical system. In case the retraining is performed too frequently, the models will rely on a small, non-representative subset of production data, therefore, they cannot provide the desired prediction accuracy. On the contrary, rare retraining can take much time due to the larger training dataset, moreover, the training data can include obsolete samples, not representing the actual state of the system.

According to the above guidelines, a random forests model was applied in the current experiments, and a retraining frequency of 5000 event samples (which equals to the data of cca. 1000 jobs) was set. These settings resulted in reasonable prediction accuracy with cca. 10% normalized root mean square error (NRMSE), with a training time around 1 sec. applying 50 decision trees. The features of the prediction include all elements of the tuple F_k, additionally, as a result of feature selection, the actual WIP (when a job is released) and the arrival time are also added to the features. These settings were resulted by offline experiments, implementing the ML models in R language [16]. This final model was deployed to be the prediction engine of the controller, as described next.

3.2. Closed-loop production controller with ML engine

As a proof-of-the-concept that real time prediction of LTs can be utilized to increase efficiency by decreasing the lateness, a simple production controller is implemented that applies the above described machine learning model in its core. The schematic architecture of the controller (prepared by following the scheme in Fig. 1) is illustrated on Fig. 3, on which colors denote the main elements of the system. The manufacturing system—replaced by its simulation model in the test case—is completed with and MES system that is responsible for streaming event data towards the controller applications (orange). This data is completed with the job features, stored separately in an ERP system (or in a similar database).

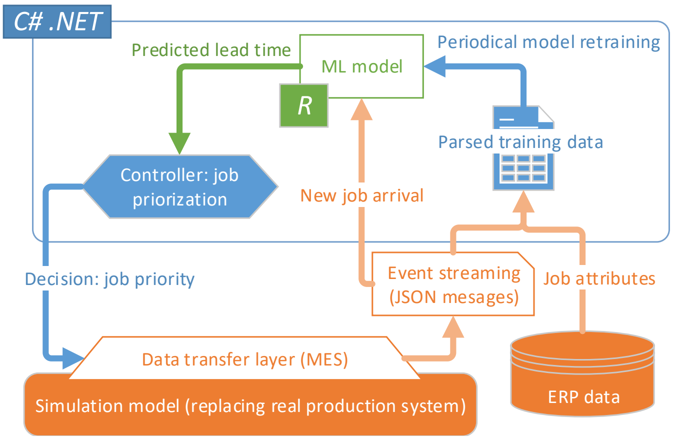

Fig. 3. Architecture of the closed-loop production controller with real time LT prediction engine.

The data processing application, and also the controller itself are implemented in C# .NET (blue), and they are responsible for receiving production data, consolidating and parsing the event stream and the static data, and also making decisions that means the job priorization in this particular case. The data parsing is done by applying a combination of R and .NET technologies, and includes the JSON parsing and matching the event data entries with the job features based on the job IDs. Similarly to the model training, this process is performed periodically in order to reduce computation times: the event stream is buffered, until the ML model is retrained, and data parsing is done in parallel with it applying the R computation engine (green). This engine is part of a periodically trained model in the R kernel, and predicts the lead times with the trained model when an event data entry about a job arrival is received. The prediction process is rather fast, therefore, the predicted lead time is immediately applied for job priorization by the controller. The result of this decision making process is a single job-priority (based on the LT prediction) that is sent back to the MES system as a JSON message, still before the job is released. This workflow enables to priorize jobs based on their predicted lead times. Various techniques and rules can be applied to do so, such an example is introduced in the following, experimental section.

4. Experimental results

The purpose of the experiments was twofold: the first and main objective was the evaluation of the proposed closed-loop control method, namely, to verify if the model can be applied in the proposed workflow and can provide sufficient predictions. The second, minor objective was the evaluation of the proposed, LT-based job priorization. As for the latter, three main methods were analyzed. First, jobs were priorized according to a FIFO rule, therefore, jobs that arrived later got lower priority. As a more realistic baseline, LTs were predicted (upon job arrival) with Little's law. In the third case, the proposed, ML-based lead time priorization was applied as it follows. Upon the arrival of a job, its lead time was predicted by applying the training ML model. In the know of the job due date and the predicted lead time, jobs with predicted late completion were given higher priority, which was proportional with the expected late (the higher the late, the higher the priority). This rule was also applied in the second scenario, the only difference was the applied prediction method (analytical vs. ML one).

Accordingly, the expectation before the experiments were the following. First (i), if the simulation as a testbed was prepared correctly, FIFO priorization should provide the worst results with the highest overall lateness Λ. Second (ii), analytical prediction need to outperform significantly the FIFO rule. If this holds, one can verify that job due-dates are correctly set, and lateness is sensitive to the lead times and the priorization. Third (iii), ML-based prediction models need to perform as good as the analytical ones, and in case they perform the best, the efficiency of the overall approach can be proven. Six different—considered to be representative—scenarios were analyzed, which differed in the job arrival rates and the random parameters (e.g. processing times and machine breakdowns), and each of them had the same horizon of seven days. The overall results are summarized in Table 1.

Table 1. Experimental results: total lateness of jobs on a 7-days horizon, expressed in hours; negative values indicate early (overall) completion.

	Λ [hours]		
Scenario [#]	FIFO	Little's law	ML
1	215.7	96.0	16.1
2	220.2	203.8	195.6
3	85.5	78.7	-39.7
4	184.4	114.0	105.0
5	7.3	-59.6	-171.8
6	80.4	2.9	-49.9

The average prediction accuracy of the ML model was similar to those obtained within the offline study (9-10% NRMSE). Due to the dynamics of the system, in some cases, a drift in the accuracy was identified. This drift was periodically terminated by the model retraining, so as recovering a reasonable overall accuracy (Fig. 4). The numerical results prove the assumption that the online prediction based control can increase the overall effectiveness by reducing the lateness of jobs. Referring back to the expectations, the scenarios were defined correctly, as (i) the FIFO priorization provided the worst results in all cases, therefore, one can derive that job priorization influences the lateness. Observing the third column of the table including the results achieved with Little's law, (ii) jobs' due date assignment policy is considered to be correct, as the online prediction with an analytical model could decrease the lateness with a better priorization than the simple FIFO rule. The most important overall result is the fact that ML-based prediction could outperform the other two methods, therefore, (iii) the results mark that it

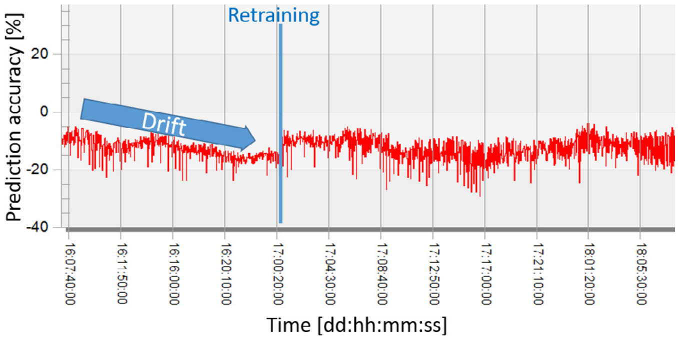

Fig. 4. Sample chart of the real-time LT predictions and the accuracy evaluation (when a job is completed), with a drift effect and the accuracy recovering with retraining. Outlier (inaccurate) predictions typically correspond to reworked items.

is worth to further investigate and improve the proposed, ML-based online production control method.

5. Conclusions and outlook

Although the results obtained during the experiments are very promising, the authors consider this work only as a step towards a sophisticated, real-time and situation-aware production control framework. As a conclusion of the experiments, one can identify that there is a potential in the application of ML methods in PPC: they can react on the dynamic changes of the production environment, while still capable of providing accurate predictions that can be directly applied in control-related decision making processes. As for the application of the method in a real manufacturing environment, candidate test environments are from the the mass customization field, e.g., from the optics industry, where on-time delivery and, therefore, manufacturing lead times are of crucial importance, and batch production is not possible.

As for the future work, major steps towards the desired control framework include the extension of considered control parameters, e.g. OEE related metrics. Besides, in case data is streamed in a higher frequency, more sophisticated, advanced data analytics tools might be required, e.g., Apache Spark™, Storm™or Flink®. These tasks ask for a more generic reference control architecture with data collection, storage (e.g. corporate data lake) and processing layers, that can be applied in various PPC environments. In this regard, more complex decisions will be also considered, including resequencing, mid-progress priorization and resource selection tasks.

Acknowledgements

The research in this paper was supported by the European Commission through the H2020 project EPIC (https://www.centre-epic.eu/) under grant No. 739592, and by the GINOP-2.3.2-15-2016-00002 grant on an ”Industry 4.0 research and innovation center of excellence”.

References

[1] Alenezi, A., Moses, S.A., Trafalis, T.B., 2008. Real-time prediction of order flowtimes using support vector regression. Computers & Operations Research 35, 3489–3503.

[2] Allen, S.T., Jankowski, M., Pathirana, P., 2015. Storm Applied: Strategies for Real-time Event Processing. 1st ed., Manning Publications Co., Greenwich, CT, USA.

[3] Bertsimas, D., Kallus, N., 2014. From predictive to prescriptive analytics. arXiv preprint arXiv:1402.5481 .

[4] Box, G.E., Jenkins, G.M., Reinsel, G.C., Ljung, G.M., 2015. Time series analysis: forecasting and control. John Wiley & Sons.

[5] Cardin, O., Trentesaux, D., Thomas, A., Castagna, P., Berger, T., El-Haouzi, H.B., 2017. Coupling predictive scheduling and reactive control in manufacturing hybrid control architectures: state of the art and future challenges. Journal of Intelligent Manufacturing 28, 1503–1517.

[6] Esposito, C., Ficco, M., Palmieri, F., Castiglione, A., 2015. A knowledge-based platform for big data analytics based on publish/subscribe services and stream processing. Knowledge-Based Systems 79, 3–17.

[7] Gyulai, D., Pfeiffer, A., Nick, G., Gallina, V., Sihn, W., Monostori, L., 2018. Lead time prediction in a flow-shop environment with analytical and machine learning approaches. IFAC Proceedings *16th IFAC Symposium on Information Control Problems in Manufacturing–IFAC INCOM 2018, Bergamo, Italy, In Print*.

[8] JSON, 2013. The JSON Data Interchange Format. Technical Report Standard ECMA-404 1st Edition / October 2013. ECMA. URL: http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf.

[9] Kwiatkowski, D., Phillips, P.C., Schmidt, P., Shin, Y., 1992. Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? Journal of econometrics 54, 159–178.

[10] Lingitz, L., Gallina, V., Ansari, F., Gyulai, D., Pfeiffer, A., Sihn, W., Monostori, L., 2018. Lead time prediction using machine learning algorithms: A case study by a semiconductor manufacturer. Procedia CIRP 72, 1051–1056.

[11] Little, J.D., 2011. OR forum–Little's law as viewed on its 50th anniversary. Operations research 59, 536–549.

[12] Monostori, L., Csáji, B.C., Kádár, B., Pfeiffer, A., Ilie-Zudor, E., Kemény, Z., Szathmári, M., 2010. Towards adaptive and digital manufacturing. Annual reviews in Control 34, 118–128.

[13] Öztürk, A., Kayalıgil, S., Özdemirel, N.E., 2006. Manufacturing lead time estimation using data mining. European Journal of Operational Research 173, 683–700.

[14] Papp, S., 2016. The Definitive Guide to Apache Flink: Next Generation Data Processing. 1st ed., Apress, Berkeley, CA, USA.

[15] 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.

[16] R Core Team, 2017. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Vienna, Austria. URL: https://www.R-project.org/.

[17] Sabuncuoglu, I., Comlekci, A., 2002. Operation-based flowtime estimation in a dynamic job shop. Omega 30, 423–442.

[18] Said, S.E., Dickey, D.A., 1984. Testing for unit roots in autoregressive-moving average models of unknown order. Biometrika 71, 599–607.

[19] Stricker, N., Kuhnle, A., Sturm, R., Friess, S., 2018. Reinforcement learning for adaptive order dispatching in the semiconductor industry. CIRP Annals 67, 511–514.

[20] Trapletti, A., Hornik, K., 2017. tseries: Time Series Analysis and Computational Finance. URL: https://CRAN.R-project.org/package=tseries. r package version 0.10-42.

[21] Wang, C., Jiang, P., 2017. Deep neural networks based order completion time prediction by using real-time job shop rfid data. Journal of Intelligent Manufacturing , 1–16.