



HAL
open science

Toward a sawmill digital shadow based on coupled simulation and supervised learning models

Sylvain Chabanet, Hind Bril El Haouzi, Philippe Thomas

► **To cite this version:**

Sylvain Chabanet, Hind Bril El Haouzi, Philippe Thomas. Toward a sawmill digital shadow based on coupled simulation and supervised learning models. 12th International Workshop on Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future, SOHOMA'22, Sep 2022, Bucharest, Romania. hal-03835261

HAL Id: hal-03835261

<https://hal.science/hal-03835261>

Submitted on 31 Oct 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Toward a sawmill digital shadow based on coupled simulation and supervised learning models

Sylvain Chabanet, Hind Bril El-Haouzi, Philippe Thomas

Abstract Digital Twins (DT) have been introduced as promising decision support tools in many different settings and to serve a variety of purposes. Many challenges are raised by their development, including an efficient usage of their computational resources to balance performance on precision, computational cost and speed. This study is, in particular, concerned with Digital Shadows (DS), a concept derived from DT, applied to sawmills sawing production units. A method to combine a computationally intensive sawmill simulation model with a machine learning model is proposed to predict set of lumbers sawed from logs. Numeric experiments are exposed, and the proposed method demonstrate improvements from 11% to 18% of the monitored couple regret from its baseline.

Key words: Digital shadow, Active Learning, Sawmill simulation

1 Introduction

The concept of digital shadows (DS) is derived from the concept of digital twin (DT) which is tracked back to NASA Apollo program, which built two identical vehicles, one sent to space missions and the other remaining on earth [2]. The objective was to monitor the twin sent to space during mission and find solutions when faced with problems. Since then, the concepts and definitions proposed for DS and DT have evolved and it remains a field of active research. A broad generally accepted definition encompassing both DS and DT concepts was proposed as "a set of digital models of physical entities usable for optimization and decision-making purposes based on real-time information" [19]. In particular, while both DS and DT are often associated with simulation, digital models based on many technologies have been propose

Sylvain Chabanet,Hind Bril El-Haouzi, Philippe Thomas
Université de Lorraine, CRAN, CNRS UMR 7039, Campus Sciences, BP 70239, Vandœuvre-lès-Nancy, Cedex, 54506, France. e-mail: {sylvain.chabanet,philippe.thomas,hind.el-haouzi}@univ-lorraine.fr

to support their respective capabilities. For example, [7] propose an hybrid DT based on physics-based simulation augmented by a machine learning (ML) model to predict acceleration responses of a structure subject to vibrations. Similarly, [12] introduce a DT based on analytical models for the performance evaluation of a manufacturing system. While the difference between DS and DT is still subject to debates, an interesting discussion is provided by [18]. DT should be able to automatically influence their physical counterpart. As such, they require an extremely precise representation and detailed models of its behavior. DS, on the other hand, are decision support tools designed to answer specific questions. It should be noticed, however, that this distinction is not always made in the scientific literature, and DS can hardly be studied independently from DT.

Due to the variety of technologies proposed to design them, DS might be better described based on their capabilities : to maintain a description of the physical counterpart (PC) based on real time or near real time data, to predict future states of the PC or compare various configurations, and to support decision making by returning instruction or optimized plans to the physical world. Such capabilities are naturally built on top of one another. Descriptive capabilities are required for predictive model to front-run production. Similarly, the output of predictive models can be fed as input to optimization models prescribing optimized plans (figure 1).

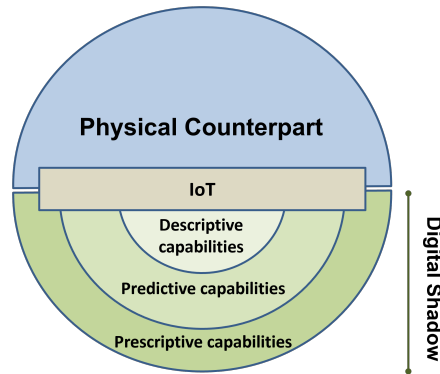


Fig. 1 The three main capabilities of a complete DS decision support tool

Digital shadows and digital twins have been studied in very different contexts, from hospitals [9] to sawmills [5]. Sawmills, in particular, might benefit greatly from DS for operational production monitoring and control. Sawing operations, the breaking of logs into lumber, is a process divergent and in co-production. From a single log, a sawmill will obtain simultaneously several pieces of lumber of varied dimensions (figure 2). In the remaining of this paper, the set of lumber obtained from a single log is named a basket of products (BoP). Many factors introduce uncertainty on the yield of the

sawing process. These factors include, for example, the heterogeneity of the shapes of logs, whose general characteristics may change from one harvesting site to another, or the emergence of automated sawlines which adjust sawing operations on a log per log basis.

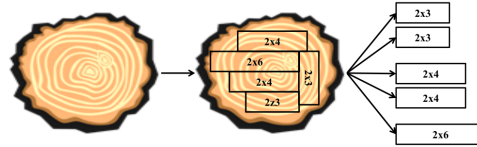


Fig. 2 Sawing a log produce several pieces of lumbers with various dimensions

All this makes it difficult to know in advance the mix of lumber that would be obtained from sawing a batch of logs. Solutions proposed in the literature to overcome this uncertainty include the use of specialized optimization methods, such as robust or stochastic optimization [23], or the usage of sawing simulators able to predict logs BoP based on information over their shape [13] or even internal structure. Information over log shape may, in particular, take the form of 3D scans, for example obtained from laser scanners which may be placed at the entrance of sawmill log yards or sawing units. General simulators able to flexibly model many sawmill configurations and predict BoP of logs based on their scans have been developed by researchers and industrial. They include, for example, Sawsim¹ or Optitek [8]. Optitek, in particular, as been used in many academic studies [14, 22]. Interestingly, both of these studies demonstrate the benefit of BoP predicted by either simulation or ML based metamodels of these simulators as input for planning decision support optimization models.

Such generic sawmill simulation tools may be great steps toward the development of flexible simulation-based sawmill DS. In particular, they would provide flexible yet precise models backing predictive capabilities. Several researchers, however, have noticed the important computational time associated with running these simulations for thousands of logs [14, 22, 16], which would impair their use for real or near real-time problems. The design of DS based on simulation and ML models used simultaneously, appears, therefore, as an interesting solution to this problem. Such DT and DS mixing data-driven methods and other simulation techniques are sometimes referred to as "hybrid" in the literature, for example by [1]. The "hybrid" qualifier refers only to the variety of models involved in the predictive capabilities of the twin. An interest of these hybrid DS is that, in particular, predictions performed by a ML model are, in general, very fast. Such models may require computationally intensive training phases, but these can be performed offline when necessary. [1], for example, develop such a model to combine a simulation

¹ <https://www.halcosoftware.com/software-1-sawsim>, Last accessed on June, 2021

model based on finite element analysis with a ML model to optimize welding operations.

Several factors complicate, indeed, the use of DS based only on data-driven, ML-based predictive models. ML models, in particular, require large datasets of labeled examples to be trained on. While in some cases these training examples may be fully obtained from the physical twin, this is complicated for sawmills by the difficulty of tracking logs and lumbers through several diverging transformation processes. In a more general setting, simulation remains necessary when historical data are not accessible from the start to train ML models, for example after changing machine settings. A second reason for favoring an hybrid DT is that ML models may, on some prediction problems and depending on the simulation technology, remain more prone to error and less explainable than these simulations.

This, however, raises the question of how to couple a simulation and ML models performing the same prediction task with their respective advantages and drawback. In this paper we propose a method to automatically decide, on a log per log basis, whether its BoP should be predicted by a ML model or a simulator.

The remaining of this study is structured as follows. Section 2 first presents in more details the hybridization problem considered. Section 3 details our proposed method, for which an experimental proof-of-concept is given in section 4. Section 5 concludes this paper and exposes future works.

2 Problem description

The general problem considered in this study assumes the existence of a stream of data items. For every individual data item X , some quantity y has to be predicted by a DS. In this study, a data item is a set of information over a specific log, and y is their BoP, but it could be generalized to any other type of stream. The existence of a DS containing at least two digital models able to make this prediction is similarly assumed. The first model is a simulation model, very precise but computationally intensive. The second is a ML model, fast but overall not as precise as a simulation. This ML model may be based on any supervised learning algorithm, for example neural network or random forest. As such, it is data-driven and requires labeled examples to be trained on. The quality of a model prediction \hat{y} is evaluated by a loss function $l(\hat{y}, y)$, so that a prediction \hat{y} closest to the real value y will lead to a lower loss. No upper bound is assumed on the number of simulations that may be run in parallel. This may be an acceptable approximation if, for example, the simulation is run on a cloud service so that the amount of computational resources which can be punctually used is far greater than what is wished to be used on the long term to maintain overall computational costs reasonable. However, an asymptotic simulation budget b is targeted, i.e, a limit of the simulation budget as time goes to infinity. It is defined here as the limit of the total accumulated simulation time divided by the time length of the stream :

$$b = \lim_{T \rightarrow +\infty} \frac{1}{T} \sum_{k=0}^{\infty} p_k \mathbf{1}_{\{a_k < T\}}. \quad (1)$$

Here, p_k is the time necessary to predict y for the k_{th} data item entering the stream, a_k its time of arrival and $\mathbf{1}_{\{\dots\}}$ the indicator function.

The objective is to propose a dispatching function $\mathbf{1}_{ToSimul(X)}$ taking value 1 if the prediction of the data item X has to be performed by the simulation model. Such a decision has to be taken immediately after arrival, without consideration for future data item. Additionally, the loss of a prediction from the ML model, $l(\hat{y}_{ML}, y)$ is, of course, unknown when the decision has to be taken and can't be used directly.

For the purpose of mathematical modeling, the following assumptions are also used. The prediction of the simulation model is supposed to be exact, i.e, $l(\hat{y}_{Simul}, y) = 0$ if $\mathbf{1}_{ToSimul(X)} = 1$. The prediction time of the ML model is neglected, i.e, $l(p_k, y) = 0$ if $\mathbf{1}_{ToSimul(X_k)} = 0$. The times of arrivals of data items are assumed to follow a Poisson process of rate μ . Simulation times are sampled from an Exponential law of mean λ . These choices make the process recording the number of simulations being run simultaneously a continuous-time markov chain process, ensure the existence of a stationary limiting distribution and allows the usage of the ergodic theorem. However, the results described in the following of this section hold in a more general case, as long as stationarity and ergodicity can be assumed.

Following this modeling, consider $\alpha = \mathbf{P}(\mathbf{1}_{ToSimul(X)} = 1)$ the fraction of the stream whose prediction is obtained from the simulation model. If the decision $\mathbf{1}_{ToSimul(X_k)}$ is taken independently from past and future arrival, the process formed by the data items for which a prediction is requested from the simulation model forms a Poisson process of rate $\mu\alpha$. Noting $N(t)$ the number of simulations running in parallel at time t , the budget b can be rewritten as:

$$b = \lim_{T \rightarrow +\infty} \frac{1}{T} \int_0^T N(t) dt, \quad (2)$$

i.e., the target budget b is also the average number of simulation running in parallel at any time. Additionally, according to the ergodic theorem, it is equal to $\mathbf{E}(N)$ under the process stationary distribution. Little's law [11] yield, in turn:

$$\mathbf{E}(N) = \mathbf{E}(p)\mu\alpha = \lambda\mu\alpha. \quad (3)$$

Hence, to target the budget b , the simulation rate α must be $\alpha = \frac{b}{\lambda\mu}$.

3 Proposal

The problem considered in this study is similar to problems detailed in the active learning (AL) literature. AL is a field from the ML literature concerned with reducing the cost associated with labeling data to train an ML model. While unlabeled collection of data items X are, in many settings, easy to

gather in huge quantity, their labels y may be costly or difficult to obtain. This is often the case when the Oracle providing the labels is, for example, a team of human experts. More precisely, AL is concerned with guiding the selection of a subset of unlabeled items that are considered the more useful to improve the performances of the ML model. It is often an iterative process where a ML model has already be initially trained on a small training set and is used to define a measure m of the utility of labeling a point. Data items maximizing this measure, or having it above some threshold are then selected from an unlabeled dataset or data stream and labeled. The ML model may be periodically retrained when a sufficient amount of new labeled data have been gathered, or when another condition is triggered such as the detection of changes in the data stream. For example, [21] propose an AL method based on a Random Forest (RF) model to detect surface defects in a stream of products.

Three main scenarios exist in the AL literature, depending on the way the active learner gains access to data :

- Pool-based active learning is concerned with cases where an unlabeled dataset may be accessed at once by the learner.
- Stream-based active learning considers cases where data items are not accessible at once but one by one from a data stream. The decision to label every item has then to be taken immediately for every data item, or for small batches of item, without considering the future of the stream.
- Membership Query Synthesis scenarios do not consider the existence of real unlabeled data. Instead, the learner may create synthetic data and request labels for these. This scenario is rarely investigated as synthetic data may be difficult to label, especially for human oracles.

All three scenarios may be of interest in the context of DS. In particular, simulation-based DS may be of great interest to label synthetic data and stream-based active learning may be used to manage real time data streams continuously collected by the twin. This study focuses on this second stream based case.

Many measures m of labeling utility have been introduced for both regression and classification tasks [10]. In particular several are based on heuristic or statistical measures of the uncertainty or confidence associated with a prediction. The method proposed in this study is to use such measures of uncertainty to design the dispatching function $\mathbf{1}_{ToSimul(X)}$, under the reasoning that such a measure may bring valuable insight over the unobservable loss $l(\hat{y}_{ML}, y)$, and that ML prediction whose confidence is considered too low should preferably be obtained from the simulation model.

In particular, consider q the quantile at level $1 - \alpha$ of $m(X)$. In general, this quantile may be estimated from observed past values from the stream. If the dispatching function is taken as $\mathbf{1}_{ToSimul(X)} = \mathbf{1}_{m(X) > q}$, then a simulation result is requested for the portion α of the stream for which the ML model is considered the least confident according to m .

The ML model used in this study is a RF model [3]. RF are machine learning models popular for their ease of use, overall good results and robustness, and have been used with success in many studies in industrial context [17, 4]. They also gave good performances in past studies comparing ML models to predict BoP of logs [14, 15]. RF are ensembles of decision tree models, where the prediction is given by $\hat{y} = \bar{t}(X) = \frac{1}{B} \sum_{i=0}^B t_i(X)$. Here, $t_i(X)$ are predictions of individual trees and B is the number of trees in the forest. Individual trees are trained on bootstrap samples selected from the whole training data set.

The measure m considered here is an estimator of the sampling variance of a random forest introduced by [20]. It evaluates how much the prediction may change if a similar model had been trained on a different dataset. This measure was used for AL for example in [6] which propose a method to predict alloys melting points. It is expressed as :

$$\sum_{i=1}^k Cov[J_i, t.(x)]^2 + \sum_{i=1}^k (\bar{t}_{-i}(x) - \bar{t}(x))^2 - \frac{ek}{B^2} \sum_{i=1}^B (t_i(x) - \bar{t}(x))^2. \quad (4)$$

Here, k is the member of data items in the training set, e the Euler number, $\bar{t}_{-i}(x)$ the average of the prediction of trees which didn't use the i^{th} data item for training, and J_i the number of time the i^{th} item was present in the bootstrap sample used to train a tree $t.$. Cov is the usual covariance estimate. Considering that [20] recommend to use forests with high number a trees, comparable with the number of data items used to train it, B is fixed to 500 in this paper experiments.

If the simulation model is precise enough, its predictions, collected along the stream may, additionally, be added to a training dataset used to periodically retrain and improve the performances of the ML model with more data.

A pseudo-code summary of the proposed solution is exposed algorithm 1.

Algorithm 1 General flow of the proposed strategy

```

Initialize LabelledSet
Initialize MLmodel on LabelledSet
for x in stream do
  Predict  $\hat{y} \leftarrow$  ML prediction
  Evaluate  $m(x)$ , the labeling utility measure
  if  $m(x) < q$  then
     $\hat{y} \leftarrow$  Simulation prediction
    add  $\hat{y}$  to LabelledSet
  end if
  if Model update condition then
    Update ML model on LabelledSet
  end if
end for

```

4 Experiments and results

This section first presents the dataset used during the experiments proposed in this study. Additional details over these experiments are also given, and the results are exposed.

4.1 Materials and method

The dataset used in the experiments described in this section originates from the Canadian forest-product industry. It contains information about 2219 wood log. In particular, both the 3D scans of every logs given, as well as a collection of six describing features. An example of such a 3D scan is presented figure 3. A scan point cloud, whose points are organized into rough ellipsoids spanning the log surface.

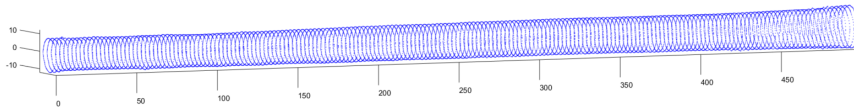


Fig. 3 3D scan of a log

The six know-how features summarizing information over the shape of logs are the length, volume, diameter at both extremities, curvature and taper (a measure of the shrinking of a log from one end to another). While the scans were used to simulate the sawing of logs and obtain their BoP, these six know-how are required by the ML model to make their own predictions. 3D scans are, indeed, unordered collection of points and the number of points varies from scan to scan. They are, therefore, unstructured data and can't be used directly by commons ML models. To summarize scans by a few descriptive features implies a loss of information that cannot be used by the ML model to make predictions. Therefore, such a ML model cannot, in particular, be expected to perform as well as a simulation model.

The BoP of every log, obtained by simulating its sawing with the simulator Optitek were also obtained. Simulation times, however, were not part of this dataset. They have been, therefore, generated randomly when required.

The sawmill simulated was able to produce up to 47 types of lumbers, characterized by their length, width and thickness. For this reason, BoP are represented as vectors of dimension 47, where the i^{th} element encodes how many pieces of lumber of type i are present in the BoP. The machine learning prediction task is, therefore, modeled as a multi-output regression problem. Considering that the measure described equation 4 is initially designed for single output regression problems, in practice a collection of 47 measures m_i

where used, one for every type of lumber, and their sum $m(X) = \sum_{i=1}^{47} m_i(X)$ was used.

To simulate a stream for experimental purpose, the dataset is ordered at random. The first 500 elements are used to initialize the RF model. Inter-arrival time in the stream were sampled from an exponential distribution with average $\lambda = 1$ and simulation time were sampled from an exponential distribution with rate $\mu = \frac{1}{10}$ so that the average simulation time is in average 10 time greater than the average inter-arrival time. Several values of the budget were tested.

At the end of the stream, the total stream time and accumulated simulation time are collected to estimate the effective simulation budget. The fraction of the stream sent to the simulation model is also collected. Two evaluation scores are additionally collected. The first one is the ML regret:

$$\frac{1}{n} \sum_{l=1}^n l(\hat{y}_{ML,l}, y_l) \quad (5)$$

with n the number of data items in the stream, y_l the label of the l^{th} arriving item, and $\hat{y}_{ML,l}$ its prediction performed by the ML model *whether it is sent to the simulator or not*. This score evaluates the performance of the ML model if it were used for every prediction.

Similarly, the coupling regret is defined as:

$$\frac{1}{n} \sum_{l=1}^n l(\hat{y}_{ML,l}, y_l)(1 - \mathbf{1}_{\{ToSimul(X_l)\}}) \quad (6)$$

and measures the performance of the hybridization. The performances of the proposed dispatching function will, additionally, be compared with the performances of a random dispatching requesting simulation depending on the result of a collection of i.i.d Bernoulli variables of parameter α . The ML model was retrained every time 100 new labeled items, or, in average, every 400 new arrivals.

To average out the impact of the exact ordering of a stream, and time sampling, this will be repeated 50 times with different random seeds.

4.2 Results

Results for three different values of the budget 1, 2.5 and 4 are presented in table 1. In particular, this means that, according to section 3, the use of the simulation model should be requested for approximately 10%, 25% and 40% of the logs from the stream to predict their BoP. This is, indeed what is observed in practice and is recorded in the column "simulated ratio". Additionally, the

respective targeted asymptotic budgets are respected, as can be seen from the observed accumulated simulation times.

budget	Method	simulation time ratio	simulated ratio	ML regret	coupling regret
1	Random method	1.01 (0.10)	0.10 (0.01)	3.25 (0.13)	2.92 (0.12)
	AL-based method	1.12 (0.07)	0.11 (0.004)	3.17 (0.10)	2.59 (0.08)
2.5	Random method	2.49 (0.17)	0.25 (0.01)	3.16 (0.09)	2.38 (0.09)
	AL-based method	2.46 (0.15)	0.25 (0.004)	3.07 (0.07)	2.03 (0.06)
4	Random method	4.00 (0.21)	0.40 (0.01)	3.09 (0.08)	1.85 (0.07)
	AL-based method	3.96 (0.17)	0.39 (0.01)	3.01 (0.06)	1.52 (0.05)

Table 1 Average and standard error over the 50 experiments of the simulation time over stream time, ratio of the stream sent to the simulator, and regrets

It is interesting to notice that the ML regret, which evaluates the performance of the ML model if it were used for all predictions, is always slightly lower for the AL-based method than for the random method. It is, indeed, the initial objective of AL methods to improve the performances of ML models by guiding the selection of data items that should be labeled rather than sampling at random. Performing Welch t-test to compare the averaged ML regrets over the various repetition of the experiment yields p-values of 0.001 for $b = 1$, 1.4×10^{-7} for $b = 2.5$ and 1.8×10^{-7} for $b = 4$. In every case, the null hypothesis of equality of the samples means can, therefore, be rejected. Despite being small, this difference is, therefore, considered statistically significant, w.r.t the repetition of the experiment. This includes randomness of the stream order, simulation time and inter-arrival time. It should be notice that the randomness from the original data collection process itself is not considered by the test. These ML regrets remain, however, high with respect to the average number of products in a basket, at 4.6, and the quadratic error obtained by a predictor predicting always the average BoP over the dataset, at 8.3. This might be partly caused by the small size of the datasets used to train these predictors.

Coupling regrets are, of course, always lower than ML regret, due to a fraction of the prediction being requested from the simulator and, therefore, considered to be exact. Once again, however, the coupling regret of the AL-based method is lower than the one of the random method, to a greater extent than what was already observed in the case of the ML regret. In particular, the regret is reduced by 11% for $b = 1$, 15% for $b = 2.5$ and 18% $b = 4$. Therefore, the sampling variance measure used in the AL-base method

appears efficient at anticipating predictions from the ML model with higher loss, and requesting predictions from the simulation model instead.

5 Conclusion

Many challenges remain in the development of DS in general, and sawmill DS in particular. The development of methods to allow an efficient usage of computational resources balancing computational cost with analytic performances appears, in particular, important for both economic and sustainability reason. To this end, the development of DS based on hybrid simulation-ML models able to benefit from the advantages of both technologies while compensating their drawback.

This study introduces a method, based on AL, to couple a computationally intensive sawmill simulator with a less precise but fast ML model to predict BoP of wood logs. The main objective is to make the best use of the computational resources allocated to the simulation model by requested simulations only for these logs for which the ML model predictions appear the least certain according to some statistical measure. During experiments, this proposed method demonstrated reductions of the simulation ML couple regret ranging from 11% to 18% depending on the simulation budget. Additionally, to use the simulation results to improve the performances of the ML model itself demonstrate a slightly better effect with the proposed AL-based method than with a random dispatching of the prediction toward the simulator.

However, several limits exist to the results presented in this paper. Firstly, more experiments need to be run to assess the impact of various parameters, in particular the average simulation time and stream arrival rate. Secondly, the respect of the simulation budget is asymptotic and based on statistical assumptions, including stationary of the data stream. This mean, in particular, that this budget may be temporarily exceeded, and increase significantly if drifts were to happen in the stream. Therefore modifications of the proposed strategy enforcing the budget to always remain between acceptable bounds appear particularly important. Additionally, many more utility measures have been proposed in the AL literature. Further works comparing different such methods will therefore be necessary.

6 Acknowledgment

The authors gratefully acknowledge the financial support of the ANR-20-THIA-0010-01 Projet LOR-AI (lorraine intelligence artificielle) and région Grand EST. We are also extremely grateful to FPInnovation who gathered and processed the dataset we are working with.

References

1. Asadi, M., Mohseni, M., Tanbakuei, M., Kashani, M.F., Smith, M.: Machine-learning-enabled digital twin of welded structures for rapid weld sequence design. In: 74th IIW on-line Assembly and International Conference (2021)

2. Boschert, S., Rosen, R.: Digital twin—the simulation aspect. In: *Mechatronic futures*, pp. 59–74. Springer, Heidelberg (2016)
3. Breiman, L.: Random forests. *Machine learning* **45**(1), pp. 5–32, Kluwer, Boston (2001)
4. Carvajal Soto, J., Tavakolizadeh, F., Gyulai, D.: An online machine learning framework for early detection of product failures in an industry 4.0 context. *International Journal of Computer Integrated Manufacturing* **32**(4-5), pp. 452–465 (2019)
5. Chabanet, S., Bril El-Haouzi, H., Morin, M., Gaudreault, J., Thomas, P.: Toward digital twins for sawmill production planning and control: Benefits, opportunities, and challenges. *International Journal of Production Research* (2022)
6. Farache, D.E., Verduzco, J.C., McClure, Z.D., Desai, S., Strachan, A.: Active learning and molecular dynamics simulations to find high melting temperature alloys. *Computational Materials Science* **209**, 111,386 (2022)
7. Gardner, P., Dal Borgo, M., Ruffini, V., Hughes, A.J., Zhu, Y., Wagg, D.J.: Towards the development of an operational digital twin. *Vibration* **3**(3), pp. 235–265 (2020)
8. Goulet, P.: *Optitek: User’s manual* (2006)
9. Karakra, A., Fontanili, F., Lamine, E., Lamothe, J.: Hospit’win: a predictive simulation-based digital twin for patients pathways in hospital. In: *2019 IEEE EMBS international conference on biomedical & health informatics, IEEE, New York* (2019)
10. Kumar, P., Gupta, A.: Active learning query strategies for classification, regression, and clustering: a survey. *Journal of Computer Science and Technology* **35**(4), pp. 913–945 (2020)
11. Little, J.D., Graves, S.C.: Little’s law. In: *Building intuition*, pp. 81–100. Springer, Heidelberg (2008)
12. Magnanini, M.C., Tolio, T.A.: A model-based digital twin to support responsive manufacturing systems. *CIRP Annals* **70**(1), pp. 353–356 (2021)
13. Maness, T.C., Norton, S.E.: Multiple period combined optimization approach to forest production planning. *Scandinavian Journal of Forest Research* **17**(5), pp. 460–471 (2002)
14. Morin, M., Gaudreault, J., Brotherton, E., Paradis, F., Rolland, A., Wery, J., Laviolette, F.: Machine learning-based models of sawmills for better wood allocation planning. *International Journal of Production Economics* **222**, 107,508 (2020)
15. Morin, M., Paradis, F., Rolland, A., Wery, J., Gaudreault, J., Laviolette, F.: Machine learning-based metamodels for sawing simulation. In: *2015 Winter Simulation Conference (WSC)*, pp. 2160–2171. IEEE, New York (2015)
16. Morneau-Pereira, M., Arabi, M., Gaudreault, J., Nourelfath, M., Ouhimmou, M.: An optimization and simulation framework for integrated tactical planning of wood harvesting operations, wood allocation and lumber production. In: *MOSIM 2014, 10eme Conférence Francophone de Modélisation, Optimisation et Simulation* (2014)
17. Ruiz, E., Ferreño, D., Cuartas, M., López, A., Arroyo, V., Gutiérrez-Solana, F.: Machine learning algorithms for the prediction of the strength of steel rods: an example of data-driven manufacturing in steelmaking. *International Journal of Computer Integrated Manufacturing* **33**(9), pp. 880–894 (2020)
18. Sapel, P., Gannouni, A., Fulterer, J., Hopmann, C., Schmitz, M., Lütticke, D., Gützlaff, A., Schuh, G.: Towards digital shadows for production planning and control in injection molding. *CIRP Journal of Manufacturing Science and Technology* **38**, pp. 243–251 (2022)
19. Savolainen, J., Knudsen, M.S.: Contrasting digital twin vision of manufacturing with the industrial reality. *International Journal of Computer Integrated Manufacturing* **35**(2), pp. 165–182 (2022)
20. Wager, S., Hastie, T., Efron, B.: Confidence intervals for random forests: The jackknife and the infinitesimal jackknife. *The Journal of Machine Learning Research* **15**(1), pp. 1625–1651 (2014)
21. Weigl, E., Heidl, W., Lughofer, E., Radauer, T., Eitzinger, C.: On improving performance of surface inspection systems by online active learning and flexible classifier updates. *Machine Vision and Applications* **27**(1), pp. 103–127 (2016)

22. Wery, J., Gaudreault, J., Thomas, A., Marier, P.: Simulation-optimisation based framework for sales and operations planning taking into account new products opportunities in a co-production context. *Computers in industry* **94**, pp. 41–51 (2018)
23. Zanjani, M.K., Nourelfath, M., Ait-Kadi, D.: Sawmill production planning under uncertainty: Modelling and solution approaches. In: *Stochastic Programming: Applications in Finance, Energy, Planning and Logistics*, pp. 347–395. World Scientific, Singapore (2013)