Toward a self-adaptive digital twin based active learning method: an application to the lumber industry

Sylvain Chabanet, Hind Bril El Haouzi, Philippe Thomas

Toward a self-adaptive digital twin based Active learning method: an application to the lumber industry

Sylvain Chabanet * **Hind Bril El-Haouzi** * **Philippe Thomas** *

* *Université de Lorraine, CNRS, CRAN, F-88000 Epinal, France (e-mail: {sylvain.chabanet, hind.el-haouzi, philippe.thomas}@univ-lorraine.fr)*

Abstract: Digital Twins (DT) is an extremely promising framework developed in the context of Industry 4.0 to facilitate the convergence of the physical and digital spaces. Numerous challenges remain, however, in terms of development, deployment, and self-adaptability of the DT faced with changes from its physical twin. Concerning this last point in particular, the set of Machine Learning (ML) methods known as Active Learning appears promising. This framework allows the DT to play an active role in the selection of the data samples used to train supervised ML models. This paper proposes a use-case inspired from the sawmill industry to illustrate the interest of these method in the presence of various changes in the flow of data gathered by the DT.

Keywords: Digital Twins, Active Learning, Artificial Intelligence, Sawmill Industry, Industry 4.0

1. INTRODUCTION

Since its introduction at the 2011 Hanover fair, Industry 4.0 has mobilized a tremendous amount of resources and attention from academics and industries alike, eager to benefit from its perceived advantages. Centered around the concept of connectivity, Industry 4.0 is based on a set of technologies and frameworks, such as the Internet of Things or Cyber Physical Production Systems. The advantages generated by these technologies include increased flexibility, reactivity, predictability and so on, allowing mass customization of production. Among this set of technologies and frameworks, the concept of Digital Twin (DT) raises important expectations. As presented in figure 1, it is sometimes introduced as a new trend in simulation technologies.

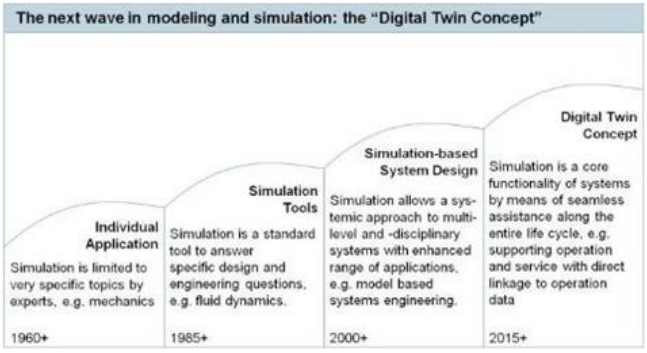

Fig. 1. New trends in simulation

The term "Digital Twin" was introduced to the public by the NASA's integrated technology roadmap under Technology Area 11: Modelling, Simulation, Information Technology and Processing (Shafto et al., 2012). No definitive definition, however, has yet been fully accepted, as many are field specific. Semeraro et al. (2021), however, propose a systematic literature review on DT and gathered up to thirty definitions that they summarize as such: *A set of* ***adaptive models*** *that emulate the behavior of a physical system in a virtual system getting* ***real time data*** *to update itself along its* ***life cycle****. The DT replicates the physical system to predict failures and opportunities for changing, to prescribe real time actions for optimizing and/or mitigating unexpected events observing and evaluating the operating profile system.*

Several points are, in particular, important in this definition. A DT gather data in real time to build a faithful, precise, and evolving representation of the physical twin. This is linked to the concept of digital shadow (Kritzinger et al., 2018) which differs from the DT in the fact that a digital shadow data flow is only automatized from the physical world toward the digital world. The DT is, therefore, completed by a set of adaptive models endowing the twin with predictive and analytic capabilities. These capabilities allow the DT to partially control the physical twin, or provide necessary decision support to human operators and managers.

Figure 2 proposes a summary outline of what may constitute a DT. It is composed of both databases and analytical or predictive models, as well as coordination mechanisms and interfaces, coordinating the different DT elements and managing both models and databases, as well as interactions with the Physical twin. Interestingly, while many authors consider the DT as a repository of *all* data and models related to the physical twins, Boschert and Rosen (2016) consider it as only a repository of the *useful* data and models. Indeed, to store unnecessary data, never used

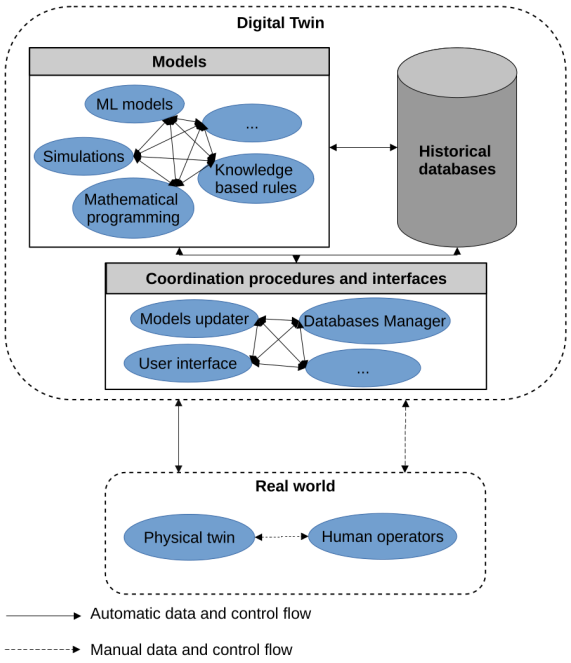

Fig. 2. Summary outline of a DT and its interactions with the real world.

by any model or process may unnecessarily monopolize storage or computational resource.

The analytical and predictive models included in the DT can be of various nature, be it advanced simulation, machine learning algorithm, or mathematical optimization models. They can, additionally, interact with each others. A simulation model could, for example, generate additional inputs for a ML predictor. Shahhosseini et al. (2021), for example, use a crop simulation model to generate additional features for a ML model predicting yield. The adaptive characteristic of these models is, however, crucial. The DT is, indeed, destined to follow its physical twin along its whole life cycle, from conception to end of life. During this time, the environment, usage, physical properties, etc, of the physical twin will change, in ways difficult to predict during the DT set-up phase. These models should, therefore, be updated, both manually and automatically, to ensure a continuously satisfactory performance level. A desirable property is, additionally, to detect and alert operators, manager, and maintenance experts when models performance risk to drop and may be inaccurate.

Among the scientific fields enabling self-updating models, ML appear of particular interest. This fields focus, indeed, on the development of models able to automatically learn from historical datasets how to generate prediction for new data samples. Use-case of ML models include, for example, predictive maintenance, where the model monitor a system state to predict future potential breakdown, or prediction about the yield of a crop or production process. A potential inconvenient of these models, is however, their requirement for a mass of labeled historical data. A labeled data point can be represented as a couple $(X, Y) \in \mathcal{X} \times \mathcal{Y}$, where X is, traditionally, a set of descriptive features. Y, called the label, is the quantity one want to predict. While the set of features X is, in general, fairly easy to obtain, generating Y may require the intervention of a costly oracle, for example a human expert or computationally intensive simulation.

Considering this fact, Active Learning (AL) has been studied during the past few decades as a framework limiting the requirement for such labeled data when training ML models, and appear extremely promising for the development of DT. This raise, however, various questions, including the behavior of AL strategies faced with changes in the Physical twin environment. The objective of the present paper is to study the behavior of an ML classifier trained following an adapted AL strategy, in the presence of abrupt and continuous changes in the nature of input data.

The remaining of this paper is structured as follows. Section 2 overviews Active Learning methods. Their interest for DT is discussed in section 3. Section 4 presents the use case, inspired from the sawmill industry. Section 5 concludes this paper.

2. ACTIVE LEARNING

The main objective of AL methods is the guided selection of a small subset of unlabeled data samples X. These unlabeled samples are then sent to an Oracle to be labeled. This labeling come generally at a cost, both in time and money, justifying the fact that the Oracle cannot be used to make each and every prediction. The labeled samples are then used to update and improve a ML predictor.

A first requirement of AL methods is to perform at least better than a random selection method. More particularly, given a predictor h_{al} trained over a dataset selected by an AL strategy, D_{al}, and a predictor h_{rd} trained over a randomly selected dataset D_{rd}, one should expect to have:

$$\mathbf{E}_{D_{al}}[\mathbf{E}_{X,Y}[s(h_{al}(X), Y)]] > \mathbf{E}_{D_{rd}}[\mathbf{E}_{X,Y}[s(h_{rd}(X), Y)]] \quad (1)$$

$\mathbf{E}_D$ represent the expected value taken over a training dataset, $\mathbf{E}_{X,Y}$ the expected value taken over new data samples (X, Y), and s a score allowing the comparison of the prediction $h_{\cdot}(X)$ given by a classifier with the real sample label Y.

Three important points, stemming from the AL strategy intended use should, in general, be considered when designing it:

- The first point to consider is the nature of the learning problem itself, i.e, what does one want to predict, and with what data. Classically, supervised learning problems are divided between classification problems and regression problems.
- The second point to consider is the *AL scenario*, or *data access scenario*. This scenario refers to the way AL selection procedure gain access to the unlabeled data samples. Three main scenarios can be identified in the literature (Kumar and Gupta, 2020). Pool-based AL refers to the case where the model has, from the start, access to the whole unlabeled database. Stream-based AL, on the other hand, refers to the case where the model has only access to data generated by a stream. The objective is to select samples without knowledge of what future samples will be

generated by the stream. Finally, membership query synthesis-based AL scenario do not have access from the start to any real dataset, but create iteratively synthetic samples labeled by the oracle. Membership query synthesis based scenario have, however, been less studied in the literature, due to the difficulty encountered by humans experts when labeling artificially generated samples (Kumar and Gupta, 2020).
- Lastly, the third point to consider is the notion of budget. This refers to the ratio, or maximal amount, of unlabeled samples which can be labeled by the Oracle. This stems from the observation that labeling is costly, in time or resource, and that a control over this cost is required.

These three points will direct the selection of both the ML predictor being trained, and AL strategy. Numerous AL strategies have been proposed over the years. A lot of them, interestingly, consider a measure of perceived certainty over the prediction yielded by the predictor in its current state. In a pool based-scenario, these samples are selected which are the most uncertain. Similarly, in a stream based scenario, this measured uncertainty over a sample can be compared to a fixed or adaptive threshold.

3. ACTIVE LEARNING FOR DT

While the problems requiring the use of such ML predictors may be varied and lead to either classification or regression problems, to discuss the choice of the data access scenario is of great interest. A model integrated to a DT will be faced with, on the one hand, new data samples being continuously gathered, and, on the other hand, changes in the physical properties or behavior of the physical twin. These changes will impact the nature of the data gathered from the physical twin sensors, and, ultimately, lower the accuracy of the model predictions. These changes in the stream of data are referred to as concepts drift in the stream-based ML literature (Webb et al., 2016) and are of various nature. In manufacturing, they may have multiple causes, from tool wear to changes in the raw material being processed. For these reason, stream-based scenario is very adapted to the case of ML models contained in a DT (Gardner et al., 2020).

Samples artificially generated by membership query synthesis based AL strategy could be seen as an interesting complement in a very specific case: when the DT contains a second model, for example a complex simulation, able to answer the same prediction task as the ML model and process artificial samples, but too computationally intensive to be used in real time. In such a case, however, strong guaranties would have to be given over the accuracy of this second model. If it were to continuously generate wrongly labeled data samples, the performances of a ML model learning from these would be impacted greatly. Some authors, indeed, argue that noise on the labels is potentially even more harmful than noise on the input features (Zhu and Wu, 2004).

The fact that many AL strategies rely on a measure of perceived certainty on the prediction of an input is, additionally, of particular interest. It may alert a decision maker, human or not, about a risk of taking decision based on these data only. Different such measures will focus on different sources of uncertainty. A first source of uncertainty is, for example, caused by data being sampled from an area of the feature space difficult to classify for an ML classification model, due, for example to classes overlap. A second important sources of uncertainty is the emergence of novelty content due to concept drift.

The literature dealing with novelty content and outliers detection has proposed numerous *novelty scores* to detect "abnormal" samples, different from samples in a "normal" dataset. Pimentel et al. (2014) classify novelty detection methods in six categories, including probabilistic and distance based. In this paper, novelty content will be considered using a distance based strategy. Such a method has, indeed, the advantage of being computable on relatively small datasets and no additional model as to be fitted to the data. Additionally, no assumption has to be made on the data distribution. This appears of particular interest, due to the challenge created by the so called sampling bias in AL applications. The statistical distribution of the AL selected training set is dependent on the AL strategy implemented, and not representative of the real data distribution.

4. USE-CASE

This section presents a simplified use case inspired from the sawmilling industry. Sawmills process raw wood logs into lumbers and other co-products. This process is divergent with co-production. In particular, several lumbers with potentially different dimensions and grades are processed from the same log. For this reason, the sawing process is more similar to a disassembly process than to an assembly process. Several factors, including the heterogeneity of the raw material, make it difficult to know in advance the set of lumbers (called in this paper a basket of products) that can be obtained from a specific log. Several numeric simulators have, however, been developed for this task. Examples of such simulator include SAWSIM, Optitek or Autosaw. Authors have, additionally, proposed the use of ML predictors trained to make this prediction (Morin et al., 2015; Martineau et al., 2021; Chabanet et al., 2021). All these models are, in fine, steps toward the construction of a sawmill DT.

4.1 Dataset

The dataset used in this section is a proprietary dataset originating from Canadian wood industry. This dataset contains data for 1207 wood logs. Each log is represented by six know-how features, based on expert knowledge from this industry. These features are, respectively, the diameters at both extremities of the log, its length, volume, curvature, and taper (a measure of its decrease in diameter). Additionally, 3D scans of the exterior shape of each log are available. These scans are point cloud, with arbitrary number of points, that span a log exterior surface.

In the absence of data from a real sawmill representing the physical twin, the 3D scans of each log had been fed to the sawing simulator Optitek to generate the baskets of products associated with each log. The simulated sawmill is able to process 19 type of lumbers which, in our dataset, are organized into 105 baskets of products. Each

basket corresponds to a class in the classification problem considered by the ML predictors. The high number of baskets w.r.t the size of the dataset makes it so that new, previously unknown classes regularly appear in the streams generated in this paper.

4.2 Stream generation

The dataset was used to generate streams containing two types of drift w.r.t the log features. A stream is, in this context, an ordering of the dataset. During experiments, samples are then considered iteratively in that order. Decision is immediately taken to add them or not to a classifier training dataset.

To simulate drift, the dataset was divided into two parts using the log length feature. One part, named cluster 1 in the following, contains 600 of the shorter logs. The second part, named cluster 2, contains the 607 remaining logs.

Abrupt drift was simulated by first iteratively selecting at random and without replacement all elements from cluster 1, to form the beginning of the stream. Then, the same procedure was done using cluster 2 to complete the stream.

Similarly, streams containing continuous drift were generated as follows: at time step i along the stream and until one of the cluster is exhausted, draw at random a Bernoulli variable U, with parameter $p = 0.2 + \frac{i}{736} * 0.7$ the values 0.2 and 0.7 are introduced here to not exhaust the first cluster too early in the stream. If $U = 0$, select a sample at random from cluster 1. Similarly, if $U = 1$, select a sample at random from cluster 2. When one of the clusters is exhausted, add all remaining elements from the other cluster at the end of the stream, ordered at random.

4.3 Evaluation scores

As usual with ML predictors, evaluation scores have to be defined to compare several models. This paper considers a variant of the F_1 score, specifically introduced by Martineau et al. (2021) to evaluate and compare ML predictors trained to predict wood logs baskets of products. This score is built on adaptations of the definitions of the number of True Positive (TP), False Positive (FP) and False Negative (FN):

- The number of True Positive is defined as the number of predicted lumbers that are effectively produced, i.e, $TP = \sum_{i=1}^{19} \min(\hat{p}_i, p_i)$, with $\hat{p}_i$ the number of lumbers of type i predicted for a specific log, and p_i the number of lumbers of type i effectively sawed from the same log.
- The number of False Positive is the number of lumbers predicted but not produced, i.e, $FP = \sum_{i=1}^{19} \min(\hat{p}_i - p_i, 0)$.
- Similarly, the number of False negative is the number of lumbers produced but not predicted, i.e, $FP = \sum_{i=1}^{19} \min(p_i - \hat{p}_i, 0)$.

The F_1 score of a predicted basket of products $\hat{p}$ is then defined as $F_1 = \frac{2 \times TP}{2 \times TP + FP + FN} \times 100$

It has to be stressed that a F_1 is computed for one prediction made for one single log, and is, here, expressed in percents. To follow the evolution of this score w.r.t the stream progress during this paper experiments, the accumulated F_1 is introduced as well. This score is initialized at 0 at the beginning of the stream. Then, each time a prediction $\hat{p}^t$ is made for the t^{th} log in the stream, its F_1 score is computed and noted F_1^t. The accumulated score is then updated as $accF_1^t = \frac{(i-1) \times accF_1^{t-1} + F_1^t}{i}$

4.4 AL sampling strategy

AL strategies based on uncertainty or novelty measures often compare the measure of the current sample generated by a stream with a threshold, and select these samples whose measure is below this threshold. This threshold may be fixed or variable (Lughofer, 2017). One inconvenient of fixed threshold is that they can be difficult to set a priory, and give little insurance in the presence of drift. Kottke et al. (2015), however, propose to use as threshold the b-quantile of the measure theoretical probabilistic distribution, estimated from past value of the stream. This ensure, in particular, that as long as the stream remains stationary, i.e, without drift, the AL strategy will select approximately $b\%$ of the stream samples, respecting a predefined budget. In fact, as the certainty measure may be expected to have a tendency to take higher values with time, as more samples are added to the training dataset, such a strategy could even be expected to slightly under-sample the stream. What is interesting with this method is it's behavior in case of abrupt drift. Indeed, in such a case, samples will suddenly arrive with small values of the certainty measure. This will lead to a temporary high sampling rate, allowing the training dataset to be automatically and quickly updated with these new concepts. In the following, b was fixed to 0.4, to sample slightly less than half the samples from the stream. In general, this budget is not easy to decide. Especially, to fix it too low might lead to poor performances from the AL selection strategy which might select too many outliers, polluting the training dataset. Overall, to fix it as high as computationally possible might appear a good rule of thumb.

The selection of a certainty measure is, often, task specific, as many of these measures are not guaranteed to perform better than random selection on every dataset. The measure proposed in this study is as follows. Let L be the current training dataset used to train the classifier h. Let x be a new log generated from the stream, and $p = (p_1, ..., p_c)$, with p_i the probability that x belong to class i, as estimated by the classifier h. Let d be the so called ICP dissimilarity introduced by Selma et al. (2018) to compare 3D log scans. This dissimilarity is a measure of how different a log x is from a second log y, and is used in this paper to detect novelty. This article propose, therefore, to use the following certainty measure m :

$$m(x) = \frac{max(p_1, ..., p_c)}{\min_{y \in L}(d(x, y) + d(y, x))} \tag{2}$$

This measure is, therefore, the ratio of the uncertainty sampling measure, which is a classic measure from the AL field, over a measure of how different the log x is from logs in L, estimated with the ICP dissimilarity.

The classifier used in this study is the Extremely Randomized Tree classifier (Geurts et al., 2006). This classifier is a

variant of the classic Random Forest Classifier, relatively fast to train as cuts are decided at random. The exact implementation used is the class ExtraTreesClassifier from the python library sklearn.

4.5 Experimental results

Experiments are designed as follows. First, a stream is generated as described above. The first 50 elements of the stream are used to train two classifiers. The first one will be subsequently upgraded using a training dataset gathered from an AL strategy, while the second one will be used for comparison purpose, and trained on a dataset gathered at random. The remaining logs are then considered one by one, in the order dictated by the stream. Their basket of products is predicted by both classifier. The exact variation of the score associated with this prediction is difficult to interpret, in particular because the baskets of products of short logs are easier to predict than the baskets of long logs. Therefore, the drift introduced in the stream makes the classification problem easier at the beginning of the stream than at the end. The difference in F_1 between the prediction made by the classifier trained on the AL dataset and the prediction made by the classifier trained on the random dataset, $\Delta F_1 = F_1^{AL} - F_1^{rd}$, with F_1^{AL} the score of the AL model and F_1^{AL} the score of the model trained on the randomly selected dataset appears, therefore, more interesting.

The certainty measure associated with the log x is then evaluated and the log is added or not to the AL dataset depending on its value. Similarly, a Bernoulli random variable with parameter $p = 0.4$ is generated, and the sample added to the random dataset if the value of this Bernoulli is 1.

A classifier is trained again every time 30 new samples have been selected from the stream by the AL or random method respectively.

To take the influence of the stream order into consideration as well as the randomness introduced by the training of the Extremely randomized tree classifier, the experiment was repeated a hundred time for both abrupt and continuous drift cases, with different random seeds.

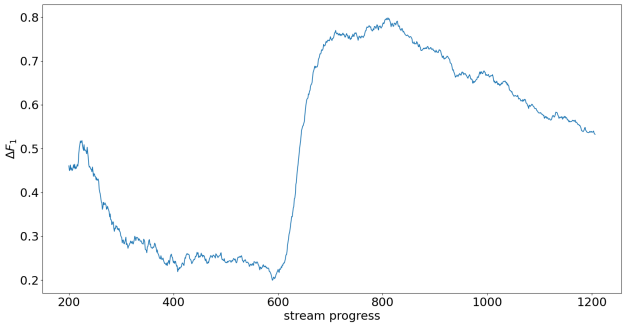

Fig. 3. Evolution of the average ΔF_1 over 100 runs of the experiment, as a function of the stream progress. The streams contain abrupt drift

Abrupt drift: Figure 3 presents the evolution of the average ΔF_1 over 100 runs of the experiment, as a function of the stream progress. A first element to notice is that this quantity is positive, even if slightly, demonstrating the advantage of the AL sampling strategy over a random sampling strategy. The last value of ΔF_1 at the end of each of the hundred streams were gathered, and a Student test was performed to test the positivity of their average. The randomness considered here is induced by the stream order and classifier training. The p-value of this test was 1.6×10^{-10}. The gain of this AL strategy is, therefore, statistically significant given our dataset.

Of particular interest is the behavior of ΔF_1 after the 600^{th} stream time step, i.e, the abrupt concept drift. A sharp increase is, indeed, observed. The AL strategy is, therefore, able to react faster than the random strategy to this drift. ΔF_1 then slowly decreases, as more samples are gathered by both strategies.

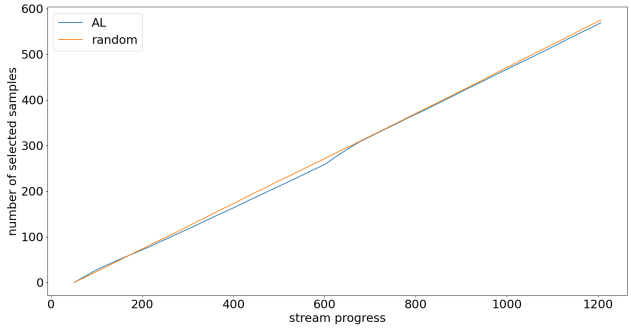

Fig. 4. Evolution of the average number of samples selected from the stream by either an AL or random strategies, over 100 runs of the experiment. The streams contain abrupt drift

The explanation for this faster adaptability of the AL strategy can be found in figure 4. This figure presents the number of samples having been gathered by either the AL or random strategy, after each stream time step. During the first half of the stream, The AL strategy selects slightly less samples than the random strategy, as may be expected for a stationary stream. just after time step 600, however, the selection speed suddenly increases, allowing the classifier trained on the AL training dataset to be quickly updated with new samples. The selection rate, then, slow down again as the stream is, once again, stationary.

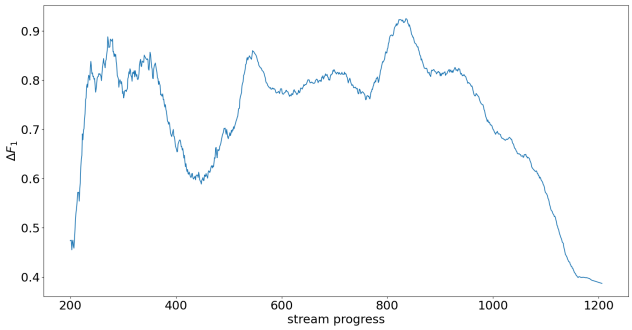

Fig. 5. Evolution of the average ΔF_1 over 100 runs of the experiment, as a function of the stream progress. The streams contain continuous drift

Continuous drift: Similarly, figure 5 presents the evolution of ΔF_1 under continuous drift. Once again, ΔF_1 is, in

average, positive. A student test was performed, yielding a p-value of 2.8×10^{-7}. By the end of the stream, however, a decreasing tendency can be observed, as both classifiers are trained on vast enough dataset that the advantage of the AL strategy decreases.

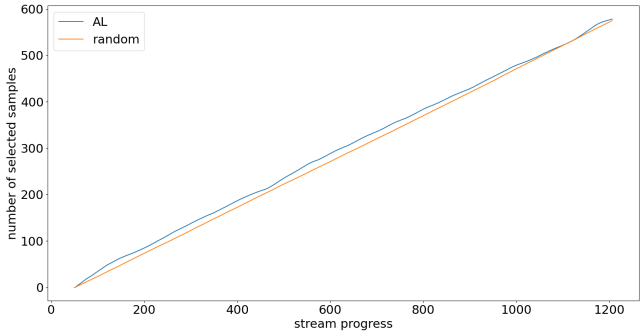

Fig. 6. Evolution of the average number of samples selected from the stream by either an AL or random strategy, over 100 runs of the experiment. The streams contain continuous drift

Figure 6 presents the evolution of the number of samples selected by either the AL or random strategy. This time, the AL strategy reacts to the constant drift by gathering a slightly larger training dataset than the random selection strategy. This difference, however, slightly decreases with the progress of the stream, with the increase in confidence of both classifiers over both sample clusters.

5. CONCLUSION

Stream based AL appear promising for the development of self-adaptive DT. As shown by this study experiments, it allows the DT to react to two types of drift on the set of features gathered from the physical twin and used as input by a ML classifier. In particular, They allow to automatically increase momentarily the number of samples being selected from the stream to update the classifier. Additionally, such a method outputs a confidence measure on the classifier prediction, which could be used to alert a manager using these to take decisions.

This work, however, only considers concept drift on the features space, also called covariate drift. Future works will, therefore, have to consider other families of drifts, such as class drift. This type of drift occurs when the class probabilities for a same set of feature change with time. Such a drift can, for example, be caused by setting changes on a production line. finally, an inconvenient of AL is that it give little theoretical that equation 1. It appear, therefore, indispensable to develop methods to switch sampling strategy when necessary.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the financial support of the ANR-20-THIA-0010-01 Projet LOR-AI (Lorraine Intellgence Artificielle) and the région Grand EST. We are also extremely grateful to FPInnovation, who gathered and processed the dataset used in this study.

REFERENCES

Boschert, S. and Rosen, R. (2016). Digital twin—the simulation aspect. In *Mechatronic futures*, 59–74. Springer.

Chabanet, S., Thomas, P., El-Haouzi, H.B., Morin, M., and Gaudreault, J. (2021). A knn approach based on icp metrics for 3d scans matching: an application to the sawing process. In *17th IFAC Symposium on Information Control Problems in Manufacturing, INCOM 2021*.

Gardner, P., Dal Borgo, M., Ruffini, V., Hughes, A.J., Zhu, Y., and Wagg, D.J. (2020). Towards the development of an operational digital twin. *Vibration*, 3(3), 235–265.

Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine learning*, 63(1), 3–42.

Kottke, D., Krempl, G., and Spiliopoulou, M. (2015). Probabilistic active learning in datastreams. In *International Symposium on Intelligent Data Analysis*, 145–157. Springer.

Kritzinger, W., Karner, M., Traar, G., Henjes, J., and Sihn, W. (2018). Digital twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine*, 51(11), 1016–1022.

Kumar, P. and Gupta, A. (2020). Active learning query strategies for classification, regression, and clustering: a survey. *Journal of Computer Science and Technology*, 35(4), 913–945.

Lughofer, E. (2017). On-line active learning: A new paradigm to improve practical useability of data stream modeling methods. *Information Sciences*, 415, 356–376.

Martineau, V., Morin, M., Gaudreault, J., Thomas, P., and El-Haouzi, H.B. (2021). Neural network architectures and feature extraction for lumber production prediction. *Proceedings of the Canadian Conference on Artificial Intelligence*.

Morin, M., Paradis, F., Rolland, A., Wery, J., Laviolette, F., and Laviolette, F. (2015). Machine learning-based metamodels for sawing simulation. In *2015 Winter Simulation Conference (WSC)*, 2160–2171. IEEE.

Pimentel, M.A., Clifton, D.A., Clifton, L., and Tarassenko, L. (2014). A review of novelty detection. *Signal Processing*, 99, 215–249.

Selma, C., El Haouzi, H.B., Thomas, P., Gaudreault, J., and Morin, M. (2018). An iterative closest point method for measuring the level of similarity of 3d log scans in wood industry. In *Service Orientation in Holonic and Multi-Agent Manufacturing*, 433–444. Springer.

Semeraro, C., Lezoche, M., Panetto, H., and Dassisti, M. (2021). Digital twin paradigm: A systematic literature review. *Computers in Industry*, 130, 103469.

Shafto, M., Conroy, M., Doyle, R., Glaessgen, E., Kemp, C., LeMoigne, J., and Wang, L. (2012). Nasa technology roadmap: modeling, simulation, information technology & processing roadmap technology area 11.

Shahhosseini, M., Hu, G., Huber, I., and Archontoulis, S.V. (2021). Coupling machine learning and crop modeling improves crop yield prediction in the us corn belt. *Scientific reports*, 11(1), 1–15.

Webb, G.I., Hyde, R., Cao, H., Nguyen, H.L., and Petitjean, F. (2016). Characterizing concept drift. *Data Mining and Knowledge Discovery*, 30(4), 964–994.

Zhu, X. and Wu, X. (2004). Class noise vs. attribute noise: A quantitative study. *Artificial intelligence review*, 22(3), 177–210.