CIRP Annals - Manufacturing Technology 70 (2021) 365–368

Contents lists available at ScienceDirect

CIRP Annals - Manufacturing Technology

journal homepage: https://www.editorialmanager.com/CIRP/default.aspx

# Semi-Double-loop machine learning based CPS approach for predictive maintenance in manufacturing system based on machine status indications

Goran D. Putnik (2)[a,b,*], Vijaya Kumar Manupati[c], Sai Krishna Pabba[d], Leonilde Varela[a,b], Francisco Ferreira[e]

[a] *Department of Production and Systems Engineering, University of Minho, Portugal*
[b] *ALGORITMI Research Centre, University of Minho, Portugal*
[c] *Department of Mechanical Engineering, National Institute of Technology (NIT), Warangal, India*
[d] *Vellore Institute of Technology, Vellore, India*
[e] *Continental Portugal S.A., Portugal*

ARTICLE INFO

*Article history:*
Available online 16 June 2021

*Keywords:*
Manufacturing system
Maintenance
Predictive maintenance

ABSTRACT

The paper presents two original and innovative contributions: 1) the model of machine learning (ML) based approach for predictive maintenance in manufacturing system based on machine status indications only, and 2) *semi*-Double-loop machine learning based intelligent Cyber-Physical System (I-CPS) architecture as a higher-level environment for ML based predictive maintenance execution. Considering only the machine status information provides rapid and very low investment-based implementation of an advanced predictive maintenance paradigm, especially important for SMEs. The model is validated in real-life situations, exploring different learning algorithms and strategies for learning maintenance predictive models. The findings show very high level of prediction accuracy.



## 1. Introduction

Applications of predictive maintenance assisted by machine learning (ML) till date involve condition monitoring by utilization of sensors for multiple parameters such as vibration, temperature, noise, pressure, speed, wear or internal damage of machine elements and tools. However, the state-of-the art modelling and applications of the predictive maintenance have some drawbacks, from which three were motivation for the model developed and presented in this paper. These are 1) it involves huge refurbishment expenditures in time and money and increase in complexity [1], 2) huge data generation for which a complex data storage and management systems are necessary [2,3], and 3) difficulties to apply in SMEs due to high costs and complexity.

The model exploits the following facts: condition monitoring needs not always to be based on a direct sensor measurement of the machine elements', or tools', conditions, but it could be predicted by the parameters that influence these condition such as the time of production, batch size, number of cycles, shift change frequency, number of setups, losses in time occurred, as these being parameters that are the major susceptible causes for conditional monitoring parameters, and especially the machine status as the main indicator for the maintenance function – being the main parameter for analysing (however, identification of the machine (resource) elements' condition based on these "external" parameters is not an objective of this paper, remaining as an issue for the future work).

Based on the history of the machine statuses, learning algorithms learn the maintenance patterns, providing the maintenance predictive models, i.e. the maintenance prognosis.

Two original and innovative contributions and the paper hypotheses are: 1) the model of ML based approach for predictive maintenance based on machine status indications only (from the reasons indicated above) and 2) *semi*-Double-loop ML based intelligent Cyber-Physical System (CPS, or I-CPS) architecture.

The model is validated by three main parameters: 1) accuracy of the prediction, 2) influence of the learning period length (implying the number of examples for learning); and 3) different strategies for learning predictive maintenance models.

## 2. Double-loop learning based Intelligent-CPS architecture for predictive maintenance

CPS is an intrinsically new control paradigm, see e.g. [4]. When the (system) control involves Artificial Intelligence (AI) and/or ML algorithms in decision making, we will designate these CPS as I-CPS.

Concerning I-CPS models, there could be conceived models with so-called *single-loop* learning – denominated $I^1$-CPS, and with *double-loop* learning – denominated $I^2$-CPS.

The $I^1$-CPS implies usual applications of the ML algorithms for learning the object system's control rule model, in our case learning of the maintenance predictive models. These learning algorithms could be called *object-learning* algorithms.

* Corresponding author.
*E-mail address:* putnikgd@dps.uminho.pt (G.D. Putnik).

https://doi.org/10.1016/j.cirp.2021.04.046

The $I^2$-CPS implies learning algorithm of the learning algorithm used in the first loop, hence *meta-learning* algorithm. The $I^2$-CPS is indicated in the definition of the CPS in [5]: "with feedback loops where physical processes affect computations and vice versa".

Fig. 1 presents the $I^2$-CPS architecture in general, and applied for predictive maintenance, instantiated for any single machine, or a group of machines, detail A in the Fig. 1.

The motivation of conceiving the "double loop" learning, i.e. the meta-learning, is from two main reasons: 1) improving the intelligence of the system, through improving learning algorithms and up to creation of new algorithms, and consequently increasing autonomy, and 2) to implement the I-CPS model in accordance with the definition by [5] cited above, which other proposed CPS models miss (for a review see [6]).

The double loop learning model, conditionally denominated as the "canonical" model, implies two separate learning algorithms, in the first and in the second loop. While the learning algorithms in the first loop are the abovementioned *object-learning* algorithms, the learning algorithm in the second loop learns about the learning algorithm in the first loop, modifying not only its parameters' values but modifying as well the parameter set and the *object-learning* algorithm's structure, generating even a totally new learning algorithm in the first loop.

As the first step towards development of the "canonical" model of the "double loop" learning, we have developed a model we call it "*semi*-double loop" learning model, as an approximation to the "canonical" model, based on selection of the learning algorithm for the "first loop", which (the selection) could be interpreted as a kind of 'primitive' learning.

I.e. in the *semi*-double-loop learning, the learning algorithm, as a 'primitive' version of learning, is reduced to 'picking' the object-learning algorithm, in an automated cycle, which outputs the model with the best predictions, from a set of different learning algorithms to be used on the object level, for a single machine or a group of machines, detail D in the Fig. 1.

In the double-loop learning model there are feedbacks, both, from the first loop to the second loop, as well as from the object-system to the second loop (providing data on the object-learning algorithm performance), as presented in Fig. 1, detail E.

Fig. 1 includes the architectural elements of the "*semi*-double loop" architecture on the right "branch" of the architecture at the digital level, as well as the "canonical" "double loop learning" architecture, on the left "branch" of the architecture.

*Double-loop learning and reinforcement learning*

Double loop learning was promoted in the context of organizational learning by Argyris C., e.g. in [7]. It could be said that the paradigm of Reinforcement Learning (RL) follows the Argyris's paradigm, representing the double-loop learning. In RL, an agent, in order to solve a problem, chooses an action $a_i$, in a sequence, from the set of available actions, in accordance with the interactions with the environment and in accordance with some policy $\pi$ (first loop). In the same time, based on the same interactions with the environment, the agent learns how to adapt the policy $\pi$ (second loop). RL is successfully implemented for a variety of applications, and in particular for predictive maintenance [8,9]. However, from the computational machine learning theory's [10] point of view, RL could be interpreted as a kind of, or specific model of, inductive inference. The "policy adaptation" is considered as the learning parameters "tuning", i.e. the learning algorithm's behaviour changes under the external input and under the policy changes, but without change of the agent's learning algorithm structure. Under this interpretation, RL could be considered, at the best, as a "*semi*-double loop" learning.

(It could be made a parallel with the difference between flexibility and reconfigurability: "Traditionally flexibility is interpreted as the ability of a system to change its behaviour without changing its configuration. Conversely reconfigurability is interpreted as the ability to change the behaviour of a system by changing its configuration." by T. Tolio in [11]).

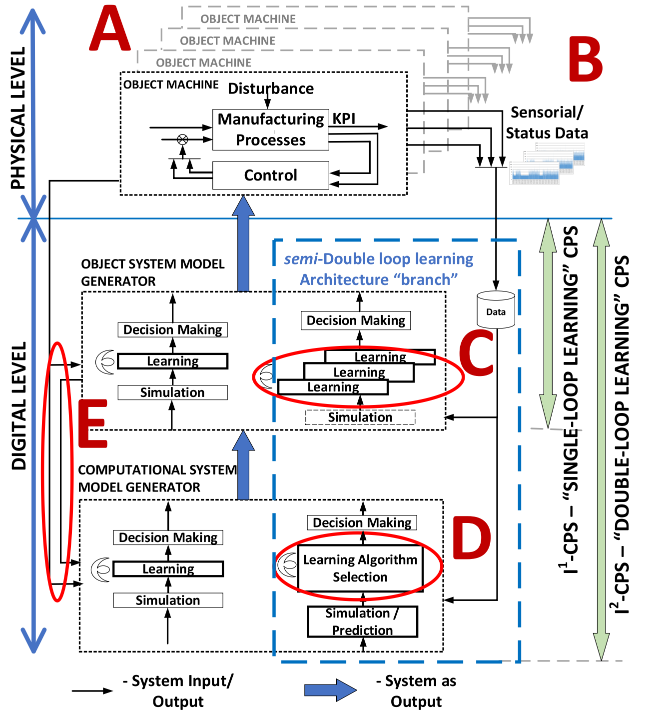


**Fig. 1.** Double-loop learning intelligent CPS ($I^2$-CPS) architecture, with semi-double-loop learning architecture "branch".

## 3. Experimentation settings

### 3.1. Demonstrator

The model of the *semi*-double-loop machine learning based CPS (*semi*-$I^2$-CPS) approach for predictive maintenance is validated in real-life industrial setting.

Data from the total of 15 machines, organized in 2 groups: 3 machines in the group 1 (id 30 to 32) performing (contextually) the 1st operation, and 12 machines in the group 2 (id 18 to 29) performing (contextually) the 2nd operation, in Fig. 2.

The model is developed for the machines in the group 2.

The data are collected for a period of 6 months, considering the variables as referred in Table 1.

A total of 30,427 batches of manufacturing data, for the machines 18 to 29, is considered.

From the collected data, detail B in the Fig. 1, during 6 months, data from the first 5 months are used for training the models for 3 different training time periods for a set of algorithms, for strategies 1, 2 and 3, ranging the training period from 3 to 5 months, and the 6th month is used for testing the performance of the trained model.

### 3.2. Learning in the 'Single-loop learning' architecture

In the single-loop learning module, 8 different learning (*object-learning*) algorithms (detail C in the Fig. 1), upon the input data presented in Table 1, were used,

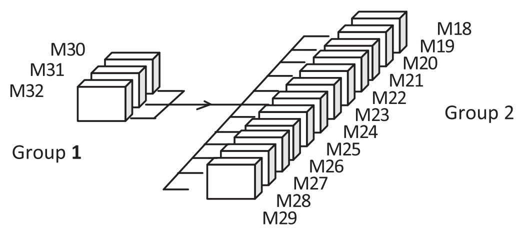


**Fig. 2.** Schematic layout of the manufacturing plant.

**Table 1**
Data variables involved in ML program.

| Input data Variables | Additional extracted feature | Output of the program |
|---|---|---|
| 1. Machine ID<br>2. Shift<br>3. Shift date<br>4. Material<br>5. Quantity<br>6. Production time<br>7. Time per piece<br>8. Time for maintenance<br>9. Setup | 1. Total work time of the machine<br>2. Total work quantity of the machine<br>3. Total number of set-ups<br>4. Total work time of the machine after previous maintenance<br>5. Total work quantity of the machine after previous maintenance<br>6. Total number of set-ups after previous maintenance | Maintenance requirement (1/0) |

*Support Vector Machine (Cubic, Quadratic with higher box constraint level), Random forest (1k & 100k learners), Naïve Bayes, Ensembled Learning (Bagged Trees and Logit Boost) and Decision Trees*
varying parameters, including hyperparameters tuning, in an automated cycle, through multiple iterations.

Concerning the algorithms evaluation, as the number of maintenance breaks after a manufacturing cycle being as low as ~5%, accuracy is not a good measure for evaluating the models (i.e. even if the models predict no maintenance required, all the time it would end up having 95% accuracy). For this reason the models are evaluated based on F1 scores (widely used for ML algorithms evaluation), see e.g. [12], for predicting requirement for maintenance and for predicting no requirement for maintenance. Additional measures, based on F1, are defined: accuracy and average F1, where

$$Accuracy = (True\ positive + True\ negative)/(True\ positive + False\ positive + True\ negative + False\ negative)$$
$$Average\_F1 = (\sum F1_{\mathrm{Mi}})/\mathrm{N}\ (i = 18 \text{ to } 29;\ N = number\ of\ machines)$$

Each particular algorithm learns the corresponded class of knowledge [13] such as: hyperplanes, decision trees, and probabilities, that separate the two classes of data, and output binary values (0/1) – for need or no need for maintenance.

The use of algorithms follows classical ML based system “pipeline”: the phase of the models training and testing, and the phase of the models deployment. In the “deployment” phase the models are used for prediction. Concerning the prediction by the models developed, the maintenance technicians know when to perform maintenance only immediately after finishing the current operation. I.e., the algorithms selected after the learning period, run after each operation for the particular machine, or the group of machines, predicting if the maintenance is required or not, before starting the new operation, and so on in cycles.

3 strategies were tested for learning the predictive model(s):

**Strategy 1 - Combined machine**: All training data for the single learning algorithm are considered as coming from one machine and if there is maintenance, we assume it to be done for one or more of the machines. *Pro: More maintenance data available to train the model and predict the maintenance break accurately. Cons: Lack of predicting the particular machine for which it has to perform maintenance*

**Strategy 2 - Multiple machines**: A single learning algorithm is used, and the training data are collected, as in Strategy 1, but the machine ID is considered as an additional input variable as the predictor. *Pro: possibility to predict the maintenance for particular machine. Cons: The usage and maintenance patterns being different for each machine, increases the difficulty in finding patterns to predict maintenance. Comparing F1-scores, strategy 2 under-performs compared with other 2 strategies.*

**Strategy 3 – Individual machine level**: Learning particular predictive model for each individual machine, as all the machines are not experiencing similar conditions, by an individual algorithm that could be different for each machine (from the set of learning algorithms referred above). *Pro: More accurate maintenance data unique to a particular machine. The technician is aware of which machine and when to perform maintenance. Cons: Complex model requiring switching of the algorithms for each machine as training data are unique to each machine*

### 3.3. Learning in the ‘semi-double learning loop’ architecture

In the *semi*-double-loop learning module, technically, the “meta-learning algorithm” in this approximation, i.e. the selection algorithm, upon multiple iterations, selects the algorithm, amongst the algorithms examined, with the highest score which will be deployed in the first loop as the model which best predicts the maintenance requirements, for each single machine or group of machines.

## 4. Experimental results

Table 2 presents the results of the 5 learning algorithms (out of 8 used) that output the predictive models with the best accuracy and the best F1 score for strategies 1 and 2, along with the average F1 of the algorithms with the highest scores per machine for strategy 3.

Table 3 lists the learning algorithms that output the best predictive model for each individual machine under Strategy 3.

Table 4 presents the overall performance of strategy 3 (average accuracy and average F1 score) with 3 different training periods. From Table 4, the 5 months of training period turns out to be the training period with the highest F1 scores.

Because of the data volume, a graphical form (diagrams) is used, Figs. 3,4.

The computational results are confirmed by the maintenance engineers in the company where the experiments took place.

**Table 2**
Results for the learning algorithms that output the predictive models under Strategies 1, 2 and 3, for the 5 months learning period.

| Learning Algorithm | Accuracy | F1 score to predict maintenance | F1 score to predict not to have maintenance |
|---|---|---|---|
| Strategy 1. – Combined Machines | | | |
| SVM (cubic) | 95.7% | 0.4906 | 0.9774 |
| RF with 1 K learners | 90.2% | 0.3838 | 0.9469 |
| RF with 100 K learners | 80.3% | 0.2807 | 0.8858 |
| Decision Trees | 90.4% | 0.4688 | 0.9713 |
| Naïve Bayes | 94.6% | 0.3077 | 0.9699 |
| Strategy 2. – Multiple Machines | | | |
| SVM (cubic) | 98.7% | 0.0000 | 0.9935 |
| RF with 1 K learners | 72.1% | 0.0543 | 0.8365 |
| RF with 100 K learners | 56.4% | 0.0423 | 0.7178 |
| Decision Trees | 80.3% | 0.0870 | 0.9829 |
| Naïve Bayes | 97.3% | 0.2609 | 0.9861 |
| Strategy 3. – Individual Machine Level (Average F1) | | | |
| Multi Algorithm learning Model | 98.76% | 0.7931 | 0.9928 |

Legend: SVM - Support Vector Machine; RF - Random Forest.

**Table 3**
Learning algorithms that output the best prediction of maintenance requirement, for each individual machine under Strategy 3, for either 3, 4 or 5 months training.

| Machine ID | Algorithm | Max F1 score |
|---|---|---|
| 18 | SVM Cubic | 0.8571 |
| 19 | SVM Quadratic with higher box constraint level | 0.8571 |
| 20 | Ensembled Bagged Trees | 0.8000 |
| 21 | SVM Cubic | 0.8000 |
| 22 | SVM Cubic | 1.0000 |
| 23 | SVM Quadratic with higher box constraint level | 0.7826 |
| 24 | SVM Cubic | 0.8000 |
| 25 | Decision Tree | 0.7273 |
| 26 | SVM Quadratic with higher box constraint level | 0.8000 |
| 27 | SVM Cubic | 0.8000 |
| 28 | Ensembled Logit Boost | 0.7619 |
| 29 | SVM Quadratic with higher box constraint level | 0.8000 |

**Table 4**
Overall performance of the strategy 3 with different training periods.

| | 3 months | 4 months | 5 months |
|---|---|---|---|
| Average Accuracy of the multi-model | 98.64% | 98.23% | 98.76% |
| Average F1 - score for predicting a maintenance occurrence | 0.7500 | 0.7200 | 0.7931 |
| Average F1 - score for predicting a not to perform maintenance | 0.9934 | 0.9931 | 0.9928 |

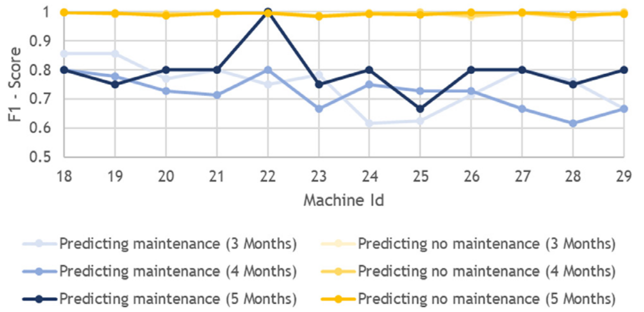


**Fig. 3.** F1 score of predicting the maintenance when required, and of predicting no maintenance required, for each machine ID for the Strategy 3 and the algorithm in Table 3.


**Fig. 4.** Overall accuracy of individual machine maintenance prediction for each machine ID for the Strategy 3 and the algorithms in Table 3.

Concerning the algorithms' response times, i.e. the computational efforts, these were at the levels of few seconds for training and testing the algorithms, and for optimizing the hyperparameters < 5 mins.

## 5. Conclusions and future work

Primarily, the main research hypothesis is validated: that the maintenance prediction, at the machine level, is possible based on machine status only. Secondly, the feasibility and rationality of the $I^2$-CPS is demonstrated as well through demonstration of the need to improve the "object learning" model.

The demonstration, based upon the system implementation in the real-life environment, showed that very good, even excellent, results could be achieved under different strategies and different learning algorithms. The main success factors are 1) selection of an adequate learning algorithm in the context, and 2) an adequate training period.

Future work: 1) investigation of new learning paradigms and algorithms, 2) continuous learning process, 3) learning "criticality index" which determines the degree of the maintenance necessity between 0 and 1, 4) new technologies for processing data, e.g. blockchain technology, 5) development of a full, *canonical* $I^2$-CPS (virtually the most challenging problem), 6) development of the model for identification of the machine elements' condition based on machine status only, and 7) improved learning performance measures.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.


## Acknowledgements

This work has been supported by FCT – Fundação para a Ciência e Tecnologia, Portugal, within the Project Scope: UIDB/00319/2020.


## References


[1] Stark R, Fresemann C, Lindow K (2019) Development and operation of Digital Twins for technical systems and services. *CIRP Ann* 68(1):129–132.
[2] Mourtzis D, Vlachou E, Milas N, Xanthopoulos N (2016) A cloud-based approach for maintenance of machine tools and equipment based on shop-floor monitoring. *Proc CIRP* 41:655–660.
[3] Wang J, Zhang L, Duan L, Gao RX (2017) A new paradigm of cloud-based predictive maintenance for intelligent manufacturing. *J Intell Manuf* 28(5):1125–1137.
[4] Monostori (2016) Cyber-physical systems in manufacturing. *CIRP Ann* 65(2): 621–641.
[5] Lee EA (2008) Cyber physical systems: design challenges. *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, IEEE, , 363–369.
[6] Putnik GD, Ferreira L, Lopes N, Putnik Z (2019) What is a cyber-physical system: definitions and models spectrum. *FME Trans* 47(4):663–674.
[7] Argyris C (2004) Double-loop learning and implementable validity. *Organizations as Knowledge Systems*, Palgrave Macmillan, London29–45.
[8] Kuhnle A, Kaiser JP, Theiß F, Stricker N, Lanza G (2020) Designing an adaptive production control system using reinforcement learning. *J Intell Manuf* 32:855–876.
[9] Kuhnle A, Jakubik J, Lanza G (2019) Reinforcement learning for opportunistic maintenance optimization. *Prod Eng* 13(1):33–41.
[10] Natarajan BK (2014) *Machine Learning: A Theoretical Approach*, Elsevier.
[11] Wiendahl H-P, ElMaraghy H, Nyhuis P, Zah MF, Wiendahl H-H, Duffie N, Brieke M (2007) Changeable manufacturing – classification, design and operation. *CIRP Ann Manuf Technol* 56(2):783–809.
[12] Joshi AV (2020) *Machine Larning and Artificial Intelligence*, Springer.
[13] Sammut C, Webb GI (2011) *Encyclopedia of Machine Learning*, Springer Science + Business Media.