Unsupervised co-training of Bayesian networks for condition prediction

Mathilde Monvoisin, Philippe Leray, Mathieu Ritou

Unsupervised co-training of Bayesian networks for condition prediction

Mathilde Monvoisin[1], **Philippe Leray**[1], **Mathieu Ritou**[1]
[1]Laboratory of Digital Sciences of Nantes (LS2N UMR CNRS 6004),
Université de Nantes, France
mathilde.monvoisin,philippe.leray,mathieu.ritou@ls2n.fr
[2]IRT Jules Verne, Nantes, France

Abstract

The objective of Smart Manufacturing is to improve productivity and competitiveness in industry, based on in-process data. It requires reliable, explainable and understandable models such as Bayesian networks for performing tasks like condition prediction. In this context, a Bayesian network can be classically learned in a supervised, unsupervised way or a semi-supervised way. Here, we are interested in how to perform the learning when the ground truth isn't included in the learning data but is observable indirectly in another related dataset. This paper introduces a fully unsupervised variation of co-training that allows to include this second dataset, with two learning strategies (split and recursive). In our experiments, we propose one simple probabilistic graphical model used for predicting the state of a machine tool from results given by several sensors, and illustrate our unsupervised co-training strategies first with benchmarks available from the UCI repository, for which 4 out of 5 datasets have best results with the recursive strategy. Finally, the recursive strategy was validated by McNemar's test as being the best strategy on a real industrial dataset.

Introduction

Smart manufacturing is a promising research area to improve productivity and competitiveness in industry (Wang et al. 2018) (Tao et al. 2018). Indeed, it is crucial to detect any system failure as early as possible to reduce maintenance costs and downtimes. This is the reason why predictive maintenance is a key issue in industry 4.0 (Gao et al. 2015). Approaches can be data-driven, model based or hybrid.

In our work, we are interested in discovering and understanding of the events leading to the damage of industrial production machine, and in predicting their failure, in a predictive maintenance perspective. In the application, we have been faced with a more general problem. Supervised learning is a classic approach for learning any predictive model when the ground truth is known. Unsupervised learning aims at learning the same model when this ground truth is unknown. In our application, we consider the unsupervised scenario, but with an additional information provided about the system state during another phase.

Smart manufacturing can make use of supervised learning in order to build a predictive model, such as a Probabilistic Neural Network to classify broken tools and good tools (Huang, Ma, and Kuo 2015) or Bayesian networks and Support Vector Machines for thermal modelling and prediction (Ramesh et al. 2003).

Improving the learning of a given model with the results of another one is one of the founding principle of the co-training. (Blum and Mitchell 1998) and (Nigam and Ghani 2000) proposed semi-supervised learning paradigm, which trains two naive Bayes classifiers respectively from two different views and lets the classifiers label some unlabeled data for each other. (Yu et al. 2011) used Bayesian undirected graphical model for co-training.

This paper introduces a fully unsupervised variation of co-training and several learning strategies are proposed, the first one to our knowledge. It includes a conditional linear Gaussian Bayesian Network structure and the learning strategies associated. The strategies were experimentally tested on 5 UCI datasets, and on a real industrial dataset dedicated to the diagnosis of machine tool. The result is the first generic framework for fully unsupervised diagnosis, which parameters are learned with co-training.

Section **problem statement** proposes a formal description of the problem. Section **unsupervised co-training** describes our unsupervised co-training framework, as well as three learning strategies. Section 6 is dedicated to the empirical evaluation of the proposal. In section 6, we describe one simple hybrid Bayesian network dedicated to condition prediction. In section 6, we will describe one first set of experiments with benchmarks available from the UCI repository, transformed for co-training task, in a controlled context where the ground truth is known. Section 6 presents an application on real industrial data, and section 6 concludes on the contribution of this paper and our perspectives of research.

Problem Statement

$\{SensA_1, ..., SensA_n\}$ is a set of continuous variables, outputs of several sensors (potentially preprocessed) that are measured every day during the "production phase" (phase A) of the machine tool. The objective is to predict $State_A$, the (discrete) state of the system, by the mean of one model $Model_A$ learned from data.

A classic approach to this problem would be to learn the parameters of $Model_A$ in a supervised way, from a dataset containing observations of the sensors

$\{SensA_1, ..., SensA_n\}$ and the ground truth about $State_A$; or in an unsupervised way, without measuring the ground truth. Let us denote D_A this dataset with the sensors information only.

In our problem, we consider the unsupervised scenario, but with an additional information D_B provided by another set of sensors about the system state during another phase (phase B). So, $\{SensB_1, ..., SensB_m\}$ is also a set of continuous variables, outputs of these other sensors (potentially preprocessed), that are also measured every day. This dataset can be used to predict $State_B$, the state of the system during phase B. States A and B of the system are assumed to be two estimations of the same underlying state of the machine. Our objective is to learn $Model_A$, with D_A, i.e. in an unsupervised way, without knowing the ground truth, but also by taking into account data D_B acquired during the second phase. As an example, in our application, D_A is the data from the process monitoring and D_B is the data collected during condition monitoring when the component signature are recorded.

As shown in a strong context by (Blum and Mitchell 1998) or a weaker one by (Balcan, Blum, and Yang 2005) for (supervised) co-training, we consider the following assumptions : (a) weak sufficiency, each of our views ($Model_A$ and $Model_B$) is at least approximately sufficient in itself to achieve good prediction, and (b) weak dependency, both views are not too highly correlated.

Unsupervised co-training

Principle

In order to solve the problem described in the previous section, we propose to "enrich" $Model_A$ with another submodel $Model_B$ dedicated to the prediction of $State_B$ from the other sensors $\{SensB_1, ..., SensB_m\}$ leading to an unsupervised co-training of both models that should agree about the state of the system.

We will consider that both models are probabilistic graphical models with continuous and discrete variables (and parameters θ_A and θ_B), and that unsupervised learning can be performed by the EM algorithm (Dempster, Laird, and Rubin 1977; McLachlan and Krishnan 2008).

This unsupervised co-training can be performed with different learning strategies, in relation to the way the equality assumption between $State_A$ and $State_B$ is envisaged.

Split learning

Instead of learning one unique model with all the sensors inputs, we propose to reduce the complexity of the parameter learning by splitting this task. i.e. learning one first submodel in an unsupervised way with the help of the EM algorithm, and then the second one enriched with the results of the first one, obtained by probabilistic inference, as described in algorithm 1. As described in figure 1, one model is then learned to "reproduce" the vector $[State_B^*]$ of the optimal outputs of the second model.

In this case, $Model_A$ takes benefit of the $Model_B$ unsupervised learning. However, in a symmetrical way, a better learning of $Model_A$ would also help for learning $Model_B$.

Recursive learning

The recursive method consists in using the previous method and iterating n_{step} times ($2n_{step}$ EM and probabilistic inference runs). Sub-model *B* is learned in an unsupervised way, enabling the supervised learning of model *A*, and then of model *B*, etc. The procedure is detailed in algorithm 2, where $P(D_A|\theta_A)$ is the likelihood of observing the data D_A for the model A, and symmetrically for model B.

For an optimal computation time, we break complexity by learning one model, and then learning the other model with the prediction of the first one, and repeating it n_{step} times. The model is learned in a sort of semi-supervised way, but without real labeled data such as in usual co-training framework.

Instead of controlling the number of steps, we can also monitor the likelihood and stop the iterations when it has stopped improving significantly.

Experiments

Model and parameter initialization

We propose to use probabilistic graphical models which are explainable models, and more especially Conditional Linear Gaussian Bayesian Networks (CLGBNs, (Lauritzen and Wermuth 1989)) that are able to deal with continuous and discrete variables.

Our unsupervised co-training framework proposes to use jointly or iteratively two models $Model_A$ and $Model_B$. We will consider here that the structure of both models is similar, and we will describe only the first one. As we are interested in applying this unsupervised co-training for condition prediction, the choice of the model used in the experiments and described in Figure 2 is inspired from the following considerations.

In an industrial system, the physical measurements (power, temperature, etc.) corresponding to good operating conditions of the system are within a limited range of values. Therefore, values outside this small and frequently observed range probably indicate problems during the manufacturing. We are defining $State_A$ as a boolean variable with {*OK*, *KO*} values. Moreover, each sensor is usually able to discriminate some intermediate states. Let us define $DiscrA_i$ the local estimation of the state provided by $SensA_i$. This variable is a discrete variable with a larger domain, for instance {*OK*, *degraded*, *KO*}. As in usual CLGBNs, we consider that the distribution of each $SensA_i$ is a Gaussian distribution conditional on $DiscrA_i$.

In order to simplify the description of the model, we will also consider that low values of $SensA_i$ usually corresponds

Algorithm 1: Split strategy

Input: D_A, D_B
Output: θ_A^*, θ_B^*
1 $\theta_B^* = \ argmax\ P(D_B|\theta_B)$
2 $[State_B^*] = \ argmax\ P([State_B]|D_B, \theta_B^*)$
3 $\theta_A^* = \ argmax\ P(D_A, [State_A] = [State_B^*]|\theta_A)$

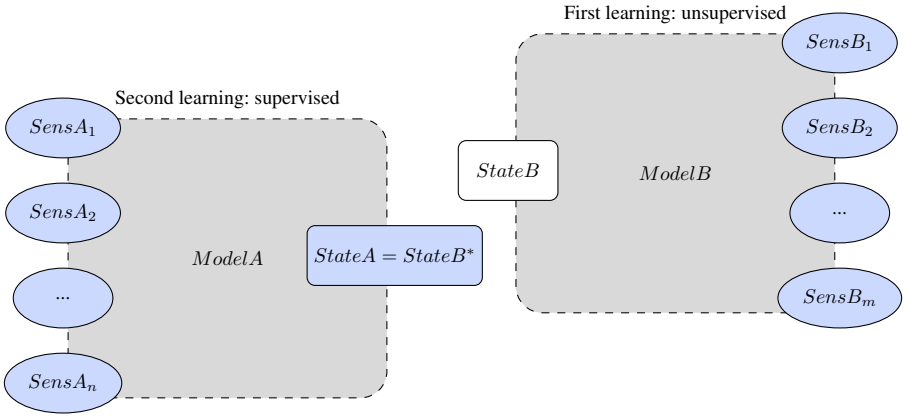

Figure 1: Unsupervised co-training by split learning strategy. Firstly, $ModelB$ is learnt in an unsupervised way, then $ModelA$ is learnt in a supervised way, by considering that $StateA$ should be the optimal prediction of $StateB$ after $ModelB$ learning. The blue nodes denote variables observed during the learning task, where the white ones represent unobserved variables.

Algorithm 2: Recursive strategy

Input: D_A, D_B
Output: θ_A^*, θ_B^*

$\theta_B^* = argmax\ P(D_B|\theta_B)$
for *i = 1 to* n_{step} **do**
  $[State_B^*] = argmax\ P([State_B]|D_B, \theta_B^*)$
  $\theta_A^* = argmax\ P(D_A, [State_A] = [State_B^*]|\theta_A)$
  $[State_A^*] = argmax\ P([State_A]|D_A, \theta_A^*)$
  $\theta_B^* = argmax\ P(D_B, [State_B] = [State_A^*]|\theta_B)$

to $State_A = OK$, and the more the sensor values increase, the more the state of the system is $degraded$ or KO.

Each conditional Gaussian distribution is initialized either by one application of EM for the joint distribution $P(SensA_i, DiscrA_i)$, or by using a classical grid initialization given in eqn 1 for a mixture of k distinct Gaussians (with i from 0 to $k-1$).

$$\mu_i = min(SensA_i) + \frac{0.5+i}{k}(max(SensA_i) - min(SensA_i)) \quad (1)$$

$$\sigma_i^2 = \frac{\sigma_{SensA_i}^2}{k}$$

We finally consider that the global state of the system is an aggregation of the local states estimated by each sensor. In a first approach that is very similar to a deterministic MAX function, we consider here that $State_A$=*KO* when it has been diagnosed as *degraded* or *KO* by at least one sensor of the model.

Experiments on UCI datasets

Dataset adaptation for co-training Several benchmarks from the UCI repository (Dua and Graff 2017) have been used. They were selected based on their similarity to our industrial context, with several numerical variables as inputs. The target variable was binarized such as the highest value corresponds to a failure if it wasn't already binary. These benchmarks are not dedicated to co-training. In order to adapt them for our task where we need two-viewed datasets, we followed a procedure described in (Ling, Du, and Zhou 2009) to split an "usual" one view dataset into two views by using an entropy based method. The two-views UCI datasets are shared on github [1] to create a public benchmark for co-training. (Ling, Du, and Zhou 2009) also introduces two measures to verify the two assumptions needed for co-training.

The sufficiency criterion δ_1 measures the fact that the two models should have sufficient information by itselves to predict the target variable. The independence criterion δ_2 measures the ability of predicting the attributes of the other dataset (given the value of the class variable). As shown in (Ling, Du, and Zhou 2009), the lower δ_1 and δ_2 are, the better the dataset is fit for co-training.

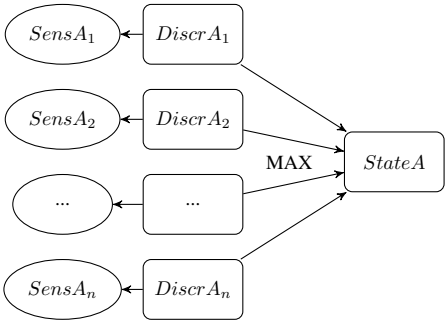

Figure 2: Conditional Linear Gaussian Bayesian Network used in our experiments for process monitoring in smart manufacturing. Circles denote continuous variables and squares discrete ones.

[1] https://github.com/MathildeMonvoisin/Co-training-benchmark

	δ_1	δ_2	n	m	N	IR
heart-statlog	0.33	0.12	7	6	270	45%
APS failure	0.11	0.39	5	5	758	36%
hydraulic stable	0.45	0.66	8	7	2206	34%
hydraulic valve	0.56	0.67	8	7	2206	67%
hydraulic leakage	0.19	0.67	8	7	2206	22%

Table 1: UCI datasets characteristics in a co-training context. δ_1 and δ_2 are the respective measures for sufficiency and independence. n and m are the number of columns of views A and B, N is the number of samples in the dataset and IR is the imbalance ratio, i.e. the percentage of the positive class.

Table 1 summarizes all the benchmarks selected from UCI repository, with their properties (total number of input variables, data size, imbalance ratio) and the sufficiency and independence measurements (δ_1 and δ_2) estimated by using a decision tree as a baseline classifier and stratified 10-fold cross-validation.

Experimental protocol The implementation of the models described in section 6 with our different learning strategies proposed in section has been performed with our library dedicated to Probabilistic Graphical Models (PILGRIM) with the help of ProBT[2] library.

We compare in these experiments our two learning strategies (Split and Recursive). As a baseline method, we also learned independently each model in an unsupervised way.

Parameters used during learning were: a threshold equal to 0.0001 on likelihood variation as stopping criterion for EM and $n_{step} = 30$ (chosen for guaranteeing convergence in all our experiments) for the recursive strategy.

The accuracy and sensitivity of each model (A and B) are estimated with 10-fold cross validation, and expressed as a value between 0 and 100%. Their average (between both models) is then considered as the global performance of each learning strategy (unsupervised, split learning and recursive learning).

Results Table 2 shows us the variation both of our metrics (accuracy and sensitivity) for each dataset.

The recursive strategy is usually better than the simple split one, with an increase in terms of accuracy and sensitivity for all datasets excepted for one dataset (hydraulic stable). The recursive co-training strategy is also more interesting than the unsupervised learning (without co-training) for three datasets (heart-statlog, APS failure and hydraulic-valve) or equivalent (for hydraulic leakage, the increase of accuracy is counterbalanced by a decrease in sensitivity).

Both co-training strategies are not efficient for the hydraulic stable benchmark and not useful for another one (hydraulic leakage). Both datasets have one high value for δ_2, showing that one of the co-training usual assumption (independence here) is not verified with also one small imbalance

[2]https://www.probayes.com/

		split vs unsup	rec. vs split	rec. vs unsup
heart-statlog	**Δ acc.**	1.67	0.56	2.22
	Δ sens.	3.12	0.21	3.33
APS failure	**Δ acc.**	0.49	0.21	0.71
	Δ sens.	11.59	1.55	13.14
hydraulic stable	**Δ acc.**	0.70	-1.82	-1.12
	Δ sens.	-0.92	-1.68	-2.60
hydraulic valve	**Δ acc.**	1.74	1.43	3.17
	Δ sens.	4.79	2.66	7.45
hydraulic leakage	**Δ acc.**	1.50	-0.06	1.44
	Δ sens.	-1.44	0.52	-0.92

Table 2: Variation of accuracy and sensitivity between unsupervised learning and the split and recursive co-training strategies.

ratio. In such imbalanced context, the first model seems to produce wrong results that are given as a ground truth for the next model. This situation can lead to a negative feedback loop and progressively produces a decrease of the global performances.

Experiments on real data from industrial use case

Data and experimental protocol This use case concerns the machining industry and the component to be diagnosed is a machine tool spindle. The industrial dataset have been collected over more than one year, aggregated at a daily level and it is unlabelled. The results obtained by a previous study (Godreau et al. 2019) which have been afterwards confirmed by an expert are considered as the ground truth.

The model A is dedicated to process monitoring during the machining phase: the four inputs are classical vibration criteria. The model B corresponds to the spindle condition monitoring: a vibration signature is performed once a day to evaluate it, with also four criteria.

Problems are uncommon in machining: from the process monitoring and the spindle monitoring datasets respectively only 7 (1.6%) and 5 (1.1%) events are considered as events that might have seriously damaged the spindle, from the previous study. Another issue in this real application is the fact that the two diagnosis don't coincide perfectly: only 3 of the 5 spindle damages are included in the 7 events detected during the process monitoring.

For this dataset, the sufficiency and independence criteria have been estimated by applying various over-sampling algorithms from the package imbalanced-learn (Lemaître, Nogueira, and Aridas 2017) because of the very low imbalance ratio ($\approx$ 1%). The results from the ADASYN, BorderlineSMOTE, KMeansSMOTE, RandomOverSampler, SMOTE and SVMSMOTE over-sampling algorithms gave results in the range $[0,02; 0,12]$ for δ_1 and $[0,34; 0,58]$ for δ_2. These values are in the range of values where co-training was efficient in our previous study on the UCI datasets.

We compare in these experiments our two co-training strategies (Split and Recursive). As a baseline method, we also learned independently each model in a unsupervised

way without the help of the other model.

The evaluation of the learning strategies is done here by two confusion matrices comparing the predictions done by our $Model_A$ and $Model_B$ to the same ground truth. For each model, we also present the classic performance indicators (precision, recall, ...) and computation time for each model *A* and *B*. In our application, the sensitivity is a very important indicator because it measures how wrong the model was when predicting an *OK*, and there can be a big impact if we ignore a *KO*, depending on what was the damage suffered.

Results Table 3 and table 4 present the confusion matrices and the other performance indicators obtained for our two co-training strategies and the baseline unsupervised learning.

The recursive method converges to performances very similar to the split one on (B), and gives increased accuracy performances on (A) and they both have a very fast learning time (less than 10s).

Table 5 presents the results of McNemar's statistical test (Dietterich 1998) where the null hypothesis considers that the predictive performances of the two classifiers are equal (with a significance level of $\alpha = 0.05$). The tests were computed using the statsmodel python library (Seabold and Perktold 2010). This table shows that the split strategy improves the spindle diagnosis results, compared to the unsupervised one. Recursive learning is globally the best strategy, by significantly outperforming all the other strategies except the split one for the spindle condition diagnosis.

The fact that the recursive strategy is not always better than the split one means that repeating the learning iterations can decrease precision and sensitivity in the process diagnosis classifier. This situation can be explained by several hypothesis: our unsupervised co-training is considering that both models should agree about their outputs, where the ground truth considered for the evaluation metrics is based on independent results for both models that disagree about some KO events. Some improvements addressing this situation are proposed in the following section.

A detailed study with a machining expert has validated a new incident in the data that was detected using the recursive strategy. This rare event has not been previously detected by other techniques that were used on this dataset.

Conclusion and future work

The paper focuses on learning a predictive model in a smart manufacturing context with an unsupervised framework where additional information is provided about the system state during another phase.

We have proposed one fully unsupervised variation of co-training framework with several learning strategies, which can be applied to various models and application fields.

These strategies have been illustrated with several benchmarks available from the UCI repository (and adapted for a co-training purpose), and have then been applied in a real application dedicated to the detection of machine tool failure when the ground truth is unknown. We have shown that our unsupervised co-training strategies can take profit from separate information in order to provide better results.

This present work can yet be extended or improved in several ways. In order to avoid the negative feedback loop observed during our experiments, the split and recursive strategies could be improved by not transferring from one sub-model learning to the second one the state predicted by the sub-model (cf. eqn 1), but the probability distribution of the state as a soft evidence, as proposed in multi-agent context (Vomlel 2004). We can also inspire ourselves from the semi-supervised co-training strategies with only a partial transfer of information between the two learning tasks, as proposed in (Nigam and Ghani 2000) where only the more confident prediction are transferred. As the sufficiency assumptions are not always met in real applications, we are also interested by extending our work with insufficient views, such as proposed for co-training in (Guo and Wang 2019).

The model structure we proposed in a smart machining context is a very simple one, used to highlight the interest of the co-training strategies. This model can be improved by taking into account more complex deterministic aggregation functions (for instance the AtLeastK operator instead of the MAX one) or probabilistic ones like NoisyMax (Srinivas 1993) or other causal independence models (Dıez and Druzdzel 2006).

Acknowledgement

Thanks are addressed to the IRT Jules Verne, French Institute in Research and Technology in Advanced Manufacturing for the PhD PERFORM program. The authors would also like to thank people who laid the foundations of this project with Philippe Leray and Mathieu Ritou, in particular Victor Godreau, Abdal Moughit Idrissi and Guillaume Ferrand, and the industrial partners for supplying the experimental data.

References

Balcan, M.-F.; Blum, A.; and Yang, K. 2005. Co-training and expansion: Towards bridging theory and practice. In *Advances in neural information processing systems*, 89–96.

Blum, A.; and Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. In *COLT' 98: Proceedings of the eleventh annual conference on Computational learning theory*, 92–100.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B* 39: 1–38.

Dietterich, T. G. 1998. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation* 10(7): 1895–1923.

Dıez, F. J.; and Druzdzel, M. J. 2006. Canonical probabilistic models for knowledge engineering. *UNED, Madrid, Spain, Technical Report CISIAD-06-01* .

Dua, D.; and Graff, C. 2017. UCI Machine Learning Repository. URL http://archive.ics.uci.edu/ml.

Gao, R.; Wang, L.; Teti, R.; Dornfeld, D.; Kumara, S.; Mori, M.; and Helu, M. 2015. Cloud-enabled prognosis for manufacturing. *CIRP Annals - Manufacturing Technology* 64(2): 749–772.

Godreau, V.; Ritou, M.; Chové, E.; Furet, B.; and Dumur, D. 2019. Continuous improvement of HSM process by data mining. *Journal of Intelligent Manufacturing* 30(7): 2781–2788.

Guo, X.; and Wang, W. 2019. Towards making co-training suffer less from insufficient views. *Frontiers of Computer Science* 13(1): 99–105.

Huang, P. B.; Ma, C.-C.; and Kuo, C.-H. 2015. A PNN self-learning tool breakage detection system in end milling operations. *Applied Soft Computing* 37: 114 – 124.

Lauritzen, S.; and Wermuth, N. 1989. Graphical Models for Associations Between Variables, some of Which are Qualitative and some Quantitative. *The Annals of Statistics* 17: 31–57.

Lemaître, G.; Nogueira, F.; and Aridas, C. K. 2017. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research* 18(17): 1–5. URL http://jmlr.org/papers/v18/16-365.

Ling, C. X.; Du, J.; and Zhou, Z.-H. 2009. When does Co-training Work in Real Data? In Theeramunkong, T.; Kijsirikul, B.; Cercone, N.; and Ho, T.-B., eds., *Advances in Knowledge Discovery and Data Mining*, 596–603. Springer Berlin Heidelberg.

McLachlan, G.; and Krishnan, T. 2008. *The EM algorithm and extensions*. Wiley series in probability and statistics. Hoboken, NJ: Wiley, 2. ed edition.

Nigam, K.; and Ghani, R. 2000. Analyzing the Effectiveness and Applicability of Co-Training. In *Proceedings of the Ninth International Conference on Information and Knowledge Management*, CIKM '00, 86–93. New York, NY, USA: Association for Computing Machinery.

Ramesh, R.; Mannan, M.; Poo, A.; and Keerthi, S. 2003. Thermal error measurement and modelling in machine tools. Part II. Hybrid Bayesian Network—support vector machine model. *International Journal of Machine Tools and Manufacture* 43(4): 405 – 419.

Seabold, S.; and Perktold, J. 2010. Statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*, 92–96.

Srinivas, S. 1993. A generalization of the noisy-or model. *UAI'93 Proceedings of the Ninth international conference on Uncertainty in artificial intelligence* 208–215.

Tao, F.; Qi, Q.; Liu, A.; and Kusiak, A. 2018. Data-driven smart manufacturing. *Journal of Manufacturing Systems* 48: 157 – 169. Special Issue on Smart Manufacturing.

Vomlel, J. 2004. Probabilistic reasoning with uncertain evidence. *Neural network world* 14: 453–466.

Wang, J.; Ma, Y.; Zhang, L.; Gao, R. X.; and Wu, D. 2018. Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems* 48: 144 – 156. Special Issue on Smart Manufacturing.

Yu, S.; Krishnapuram, B.; Rosales, R.; and Rao, R. B. 2011. Bayesian Co-Training. *Journal of Machine Learning Research* 12(80): 2649–2680.

	A				B			
	FP	FN	TP	TN	FP	FN	TP	TN
Independent unsupervised learning	1	15	414	6	0	38	393	5
Unsupervised co-training (split)	1	15	414	6	0	9	422	5
Unsupervised co-training (recursive)	3	2	427	4	0	8	423	5

Table 3: Process diagnosis (A) and spindle condition diagnostic (B) confusion matrices for several unsupervised co-training strategies compared with a baseline independent learning of both models.

	A: Process diagnosis				B: Cond. monitoring				
	Prec.	Sens.	Rec.	Acc.	Prec.	Sens.	Rec.	Acc.	Time (s)
Unsupervised	99.7%	85.7 %	96.5%	96.3%	100%	100%	91.2%	91.3%	2.1
Split strategy	99.8%	85.7 %	96.5%	96.3%	100%	100%	97.9%	97.9%	2.8
Rec. strategy	99.3%	57.1%	99.5%	98.9%	100%	100%	98.1%	98.2%	9.8

Table 4: Process diagnosis (A) and spindle condition (B) performance indicators (precision, sensitivity, recall, accuracy and computation time) for the split and recursive (rec.) unsupervised co-training strategies, compared with a baseline independent unsupervised learning of both models.

	A : Process diagnostic			B : Spindle Condition Maintenance		
	unsupervised	split	recursive	unsupervised	split	recursive
unsupervised		=	$\neq$		$\neq$	$\neq$
split			$\neq$			=

Table 5: Results of McNemar's statistical test where the null hypothesis considers that the predictive performances of a pair of classifiers are equal (=) with a significance level of $\alpha = 0.05$. The classifiers were learned with our unsupervised co-training strategies or with a baseline independent unsupervised learning of both models.