Contribution à l'apprentissage de Modèles de Markov Cachés pour l'estimation de l'état de santé de systèmes industriels

Nesrine Keltoum Khodja, Florent Duculty, Stéphane Begot, Manuel Avila

Institut pour la Maîtrise des Risques
Sûreté de Fonctionnement - Management - Cindyniques

Contribution à l'apprentissage de Modèles de Markov Cachés pour l'estimation de l'état de santé de systèmes industriels

Contribution to Hidden Markov Models learning for estimating the state of health of industrial systems

Nesrine Keltoum KHODJA, Florent DUCULTY, Stéphane BEGOT, Manuel AVILA.
Laboratoire PRISME, pôle IRAuS, Axe automatique
IUT de l'Indre
2 Av. François MITTERRAND
36000 Châteauroux
nesrine.khodja@univ-orleans.fr, florent.duculty@univ-orleans.fr, stephane.begot@univ-orleans.fr, manuel.avila@univ-orleans.fr.

***Résumé*— Nous proposons dans cet article une approche d'identification du niveau de dégradation d'un système industriel. Pour cela, nous utilisons des Modèles de Markov Cachés (MMCs) comme outil de pronostic. Dans cette étude, nous nous sommes focalisés sur la mise en compétition de plusieurs modèles. Une nouvelle stratégie d'apprentissage a été développée. Cette dernière consiste à apprendre le « contraire » des séquences (ou formes) que nous cherchons à identifier. Par la suite, nous cherchons le non-modèle qui ressemble le moins à la séquence à identifier.**

Mots-clefs—Modèle de Markov Caché (MMC), apprentissage, non-classe, durée de vie résiduelle (RUL), pronostic.

***Abstract*— We propose in this article an approach for identifying the degradation level of an industrial system. For this we use Hidden Markov Models (HMM) as a prognostic tool. In this study, we focused on the competition of several models. A new learning strategy has been developed. The latter consists in learning the "opposite" of the sequences (or forms) that one seeks to identify. Subsequently, we look for the non-model that least resembles the sequence to be identified.**

Keywords — Hidden Markov Model (HMM), Training, Non Model, RUL, prognostic.

I. INTRODUCTION

Dans le contexte actuel de compétitivité économique, l'un des plus grands enjeux dans la gestion des systèmes industriels est de maintenir ces systèmes dans un état de fonctionnement sûr. Cela permet de garantir un niveau de fiabilité, de disponibilité et de sécurité tout en assurant la maîtrise du coût global. En effet, le coût de maintenance représente une part importante du coût opérationnel global des systèmes de fabrication ou de production. Dès lors, déterminer la durée de vie résiduelle RUL (de l'anglais Remaining Useful Life) (utilisée pour prédire le temps disponible avant la défaillance d'un composant) est devenu incontournable. Les méthodes de pronostic de défaillance sont utilisées pour développer des stratégies de maintenance prédictive en donnant des indicateurs fondamentaux pour éviter les pannes graves des systèmes industriels [1]. Ces méthodes ont largement évolué. La plupart des chercheurs se sont consacrés à améliorer la précision des résultats de prédiction et ont étudié de nombreuses méthodes efficaces à partir pour déterminer la RUL. Pour prédire de grandes quantités de données, des méthodes de modélisation sont fondées sur l'apprentissage automatique. Parmi ces méthodes on trouve principalement les réseaux de neurones (NN) [2], la régression vectorielle de soutien (SVR) [3] ou encore l'utilisation des machines à vecteurs de support (SVM) [4]. D'autres méthodes de pronostic sont apparues ces dernières décennies, comme l'utilisation des Modèles de Markov Cachés (MMCs). Ces derniers ont été appliqués avec succès sur le terrain de l'identification. En raison de leurs puissantes capacités à reconnaître des formes, de nombreux chercheurs les ont appliqués dans de nombreux domaines : la reconnaissance de caractères, la reconnaissance vocale, la reconnaissance faciale [5], la caractérisation dans la médecine (la toux) [6], la reconnaissance du comportement et enfin le domaine qui nous intéresse dans cet article : le diagnostic et le pronostic des défauts. Les premiers travaux dans ce domaine ont été axés sur la surveillance et le diagnostic des processus. Dans ces domaines, on peut citer : l'identification des types d'accidents dans une centrale nucléaire, la surveillance de la boîte de vitesses d'hélicoptère

[7], la surveillance de l'usure d'un outil d'usinage [8], le diagnostic des défauts sur un rotor [9] et la détection ainsi que la reconnaissance de défauts dans les machines tournantes.
Pour le pronostic, les premiers travaux fondés sur les MMCs sont de Chinnam et Baruah [10,11]. L'objectif était d'estimer en temps réel la RUL sur un outil d'usinage. Par la suite, les MMCs ont été utilisés dans une approche de diagnostic et de pronostic pour estimer la RUL sur des roulements mécaniques [12]. Depuis, les MMCs ont été considérablement utilisés dans des approches différentes [13,14,15], comme dans les travaux de Aggab et al [16] qui proposent une approche d'identification du niveau de dégradation d'un système. L'objectif est de proposer une démarche de modélisation permettant de déterminer le niveau de dégradation atteint par le système. La démarche proposée consiste à utiliser le formalisme de Modèle de Markov Caché (MMC) multi-flux sur des séquences obtenues avec un traitement en logique floue sur les résidus.

Dans cette étude, nous nous sommes focalisés, en particulier, sur la mise en compétition de plusieurs MMCs. Une nouvelle stratégie d'apprentissage a été développée. Cette stratégie s'inspire des principes de non-occurrence (NBA) [17]. Cet article est organisé comme suit : Dans la section II, nous faisons un rappel sur les MMCs. La section III est consacrée à la méthode proposée. La section IV présente les résultats de la méthode, une discussion détaillée sur ces résultats sera présentée dans la séction V. Enfin, la conclusion et les perspectives sont présentées dans la section VI.

II. MODELE DE MARCOV CACHE (MMC)

Les MMCs ont montré leur efficacité pour la classification de séquences d'évènements (ou de symboles) dans différents domaines (paroles, écrit…). Ils ne réclament que peu de connaissances a priori sur le système, à condition de disposer de suffisamment de données d'apprentissage. À l'heure actuelle, de nombreux systèmes d'évènements séquentiels sont construits sur la base de MMC. Un grand nombre de raffinements leur ont été apportés, en particulier, pour résoudre les problèmes de systèmes complexes. L'objectif de cette section n'est pas de présenter de façon détaillée les MMCs mais d'une façon globale. Pour les lecteurs intéressés pour en savoir plus, nous vous recommandons de lire le document L. R. Rabiner [18]. Dans cet article, nous utilisons la même notation pour les modèles.

A. Définition des MMCs et de ses paramètres

Un MMC à N états et M symboles observables peut se définir comme suit :

$$\lambda = (\pi, A, B) \quad (1)$$

- Avec A : la matrice de transition. Il s'agit d'une matrice N × N dont la somme des lignes est égale à 1 (2). Chaque élément de la matrice a_{ij} correspond à la probabilité de passage d'un état i à un état j.

$$\sum_j a_{ij} = 1, \quad \forall i, j \in [1, N] \quad (2)$$

$$\begin{pmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{1N} & \cdots & a_{NN} \end{pmatrix} \quad (3)$$

- B : la matrice de probabilité d'observation C'est une matrice N × M. Chaque élément de la matrice bj(k) correspond à la probabilité d'observer le symbole k sachant un état j.

$$\sum_k b_j(k) = 1 \ \forall j \in [1, N] \text{ et } k \in [1, M] \quad (4)$$

$$B = \begin{pmatrix} b_1(1) & b_1(2) & \dots & b_1(M) \\ b_2(1) & b_2(2) & \dots & b_2(M) \\ b_N(1) & b_N(2) & \dots & b_N(M) \end{pmatrix} \quad (5)$$

- π: le vecteur de probabilités initiales. Probabilité de débuter la chaîne avec l'un des états i.

$$\sum_i \pi_i = 1, \forall i \in [1, N] \quad (6)$$

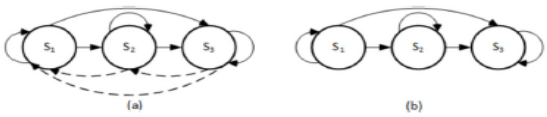
$$\pi = [\pi_1, \pi_2, \dots, \pi_N] \quad (7)$$

- On notera O = $\{O_1, O_2, \dots, O_T\}$ une séquence d'observations de longueur T produite par une séquence d'états Q = $\{q_1, q_2, \dots, q_T\}$.

B. Type de MMC

En fonction de l'usage des modèles, on peut utiliser différentes topologies. Par exemple, le modèle ergodique (Fig. 1 (a)) ou le modèle gauche-droite (Fig. 1 (b)). Le modèle ergodique est sans contrainte : toutes les transitions, d'un état vers un autre, sont possibles. Le modèle gauche-droite est un modèle « orienté » résultant de la mise à zéro de certaines transitions a_{ij} . Ce modèle gauche-droite est particulièrement adapté aux problèmes tels que les traitements de la parole ou de l'écrit.

Fig. 1 .Type de MMC modèle ergodique (a) modèle gauche-droite (b) [7]



C. Trois problèmes fondamentaux pour les MMCs

Les MMCs trouvent leur intérêt dans la résolution des trois problèmes fondamentaux que sont l'évaluation, le décodage et l'apprentissage [18].

- **Évaluation** : étant donné un MMC, il est essentiel de déterminer la probabilité qu'une séquence observée soit générée par ce modèle. Les algorithmes Forward et Backward sont utilisés pour cette tâche. Cependant, dans le cadre d'un dispositif de suivi de l'état de santé d'un système, seule la variable Forward sera utilisée, la variable Backward étant calculée depuis la fin de la

chaîne. Cela suppose que l'on a déjà atteint l'état de panne.

- **Décodage**: les états d'un MMC ne sont pas observables. Le problème du décodage consiste à chercher la séquence d'états Q la plus probable, étant donné une séquence d'observations O. L'algorithme de Viterbi est généralement utilisé pour résoudre ce problème.

- **Apprentissage** : le but est d'ajuster les paramètres du MMC, afin de maximiser la probabilité d'observation P(O|λ).

L'algorithme Baum Welch est souvent utilisé pour cette tâche. En débutant le processus d'apprentissage avec un ensemble de paramètres (π, A, B) tiré aléatoirement, un processus récursif va permettre de réestimer les paramètres du modèle pour converger vers un modèle λ^* qui maximise la vraisemblance des séquences O du corpus d'apprentissage sachant le modèle P(O|λ).

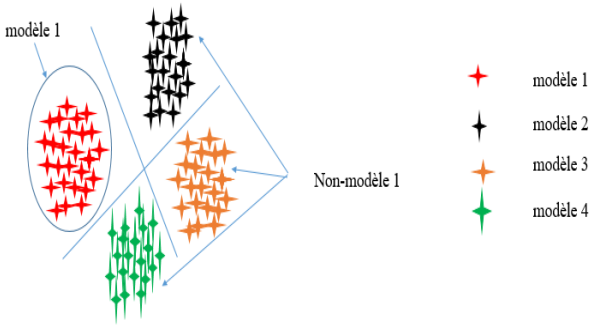
L'apprentissage peut également s'effectuer avec l'algorithme de Viterbi. Dans ce cas, seule la séquence d'états la plus probable est prise en compte dans le processus récursif de réestimation des paramètres, contrairement à l'algorithme de Baum Welch qui considère l'ensemble des séquences d'états possibles.

D'autres variantes ont été développées [19], comme les algorithmes génétiques ou des fourmis. L'idée générale, pour éviter de converger vers un minima local, est de remettre en question l'apprentissage en cours pour repartir avec de nouveaux paramètres obtenus par assemblage de différents paramètres d'apprentissage.

III. MÉTHODE PROPOSÉE

Ce travail se positionne dans le cadre d'application visant à identifier une classe parmi plusieurs possibles. Ces classes sont par exemple les différents modes de défaillance possibles auxquels s'ajouterait le mode sans défaillance. Durant la phase d'apprentissage, il faut disposer de données étiquetées pour chacune des classes.
Par exemple, sur la (Fig. 2), nous illustrons une situation avec quatre classes pour lesquelles quatre modèles doivent être estimés. Avec les méthodes « classiques » d'apprentissage, chaque modèle est estimé en utilisant les échantillons propres à sa classe.

Fig. 2. Principe des modèles et non-modèles



Dans notre étude, nous souhaitons pouvoir utiliser les informations potentiellement contenues dans les autres classes dans ce que nous avons nommé les non-modèles. Comme le montre la (Fig. 2), le pendant du modèle 1 serait l'association des échantillons des autres classes pour estimer le non-modèle 1.

Dans un premier temps, notre objectif est de vérifier que les non-modèles ont la capacité à discriminer les classes (reconnaître ce qu'ils sont en apprenant ce qu'ils ne sont pas).

Dans un second temps, l'objectif sera d'intégrer l'information contenue dans les non-modèles pour améliorer l'apprentissage des modèles.

A. Modèles de synthèse

Afin de vérifier nos hypothèses, nous avons décidé de produire des données synthétiques en utilisant une topologie de modèle semblable à un système réel déjà étudié qui produit des évènements ou observations similaires à un véritable système industriel. Cette architecture avait été jugée la plus proche du système réel dans les travaux de Robles [20].

Une architecture gauche-droite de MMC a été utilisée pour la production des données. Quatre jeux de paramètres ont été choisis aléatoirement pour produire des séquences de données et alimenter des bases de données. Les étapes de notre méthode se résument comme suit :

Etape 1 : Tirage aléatoire des modèles

La première étape consiste à générer quatre modèles λ_i avec Matlab. Ces modèles représentent trois modes de défaillance et un mode sans défaut.

Etape 2 : Production des données

Des observations simulées ont été générées à partir de ces quatre modèles, en définissant un nombre de séquences d'observations Q, de longueur T.

Etape 3 : Corpus d'apprentissage et de test

Ces observations sont divisées en deux corpus : les données d'apprentissage et celles de test. Nous avons opté pour 2/3 des données utilisées pour l'apprentissage des modèles et 1/3 pour les données tests.

Etape 4 : Phase d'apprentissage

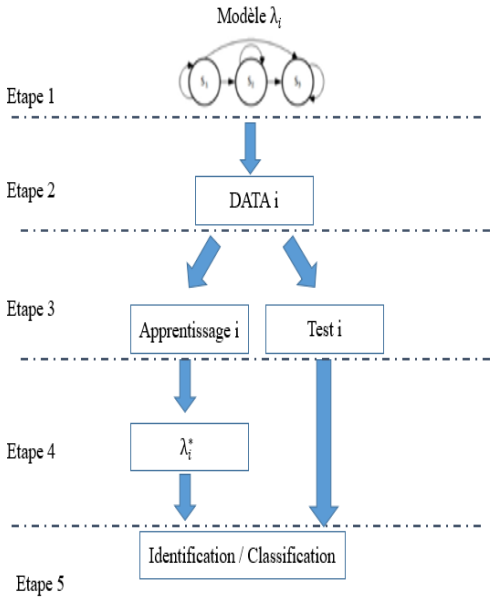
Pour la phase d'apprentissage, nous avons utilisé les algorithmes de Baum-Welch et de Viterbi pour la réestimation des modèles λ_i^*.

Etape 5 : Phase d'identification et de classification

Afin de vérifier la capacité des modèles à identifier chaque classe, nous avons employé le maximum de vraisemblance.

Les cinq étapes sont présentées dans la figure ci-dessous (Fig. 3).

Fig. 3. Les cinq étapes de notre méthode

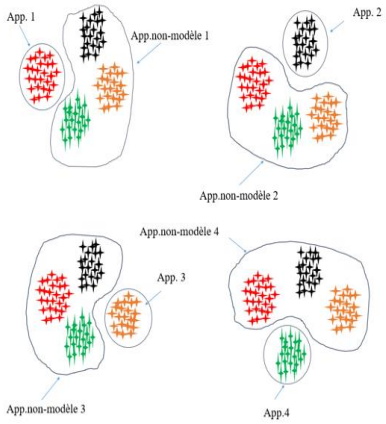


Les résultats de cette classification fondée sur l'approche classique nous permettront de disposer de taux de reconnaissance de référence pour la suite de l'étude.

B. *Discrimination par non-modèle*

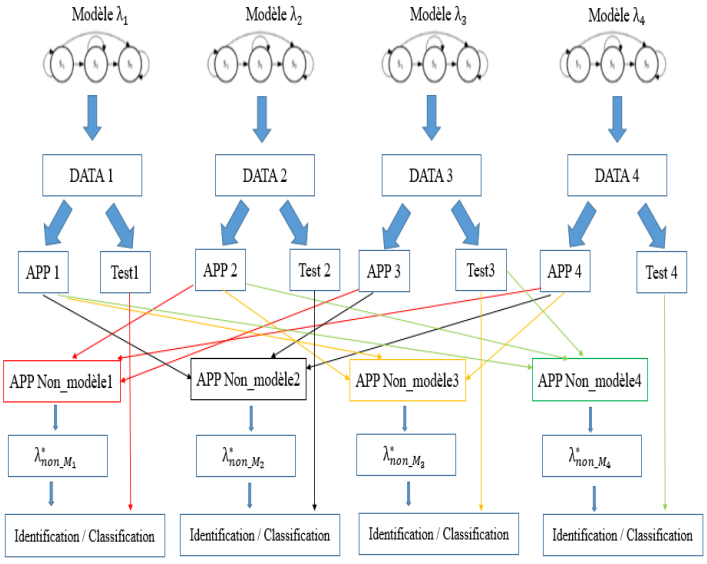
Dans la phase suivante de notre étude, nous avons considéré les échantillons des autres classes pour apprendre ce que nous avons qualifié de non-modèle (ce que l'on n'est pas) (Fig. 4). Les données utilisées pour l'apprentissage (App.) des non-modèles ont été constituées de la façon suivante :

Fig. 4. Illustration des corpus d'apprentissage des non-classes



Dans la phase de test visant à identifier les échantillons inconnus, nous calculons la vraisemblance de chaque séquence avec chacun des non-modèles qui a été estimé. Puisque nous avons appris le « contraire » des séquences (ou formes) que l'on cherche à identifier, nous cherchons la non-classe qui ressemble le moins à la séquence à identifier. Le non-modèle qui fournit la vraisemblance la plus petite est donc celui qui permet d'identifier la non-classe. La figure ci-dessous résume la méthode (Fig. 5).

Fig. 5. Méthode d'identification fondée sur les non-modèles



IV. RÉSULTATS

Après avoir défini la méthodologie, nous présentons les résultats obtenus par la modélisation « classique » qui nous servira de référence pour quantifier la capacité discriminante des non-modèles. Pour éviter de juger une situation qui pourrait être anecdotique, liée au tirage aléatoire des

paramètres des modèles, les tests ont été répétés plusieurs fois. Ce sont les moyennes des résultats obtenus qui sont fournies.

A. Résultat pour les modèles de synthèse

Nous avons généré quatre modèles. La topologie de chaque modèle est de type gauche-droite à quatre états avec émission de 16 symboles différents. La longueur des séquences O_T est de 20 observations. Chaque base de données contient 500 séquences, les corpus d'apprentissage contiennent 350 séquences de longueur 20 tandis que les corpus de tests contiennent 150 séquences de même longueur.

Dans un premier temps , nous avons effectué l'apprentissage à l'aide de l'algorithme Baum-Welch. Ensuite, nous avons calculé la vraisemblance de chaque séquence avec chacun des modèles qui ont été estimés. Pour la phase d'identification, nous avons calculé le maximum de vraisemblance.

Le résultat est présenté sous forme de matrice de confusion (qui mettra en valeur les prédictions correctes et incorrectes). C'est une moyenne sur dix tests effectués en pourcentage (%) (voir TABLE. I).

TABLE I. MATRICE DE CONFUSION POUR LES MODÈLES (APPRENTIQQAGE BAUM-WELCH) EN POURCENTAGE (%)

	Classe 1	**Classe 2**	**Classe 3**	**Classe 4**
Test1	**87**	3,8	5	4,2
Test 2	3,86	**84,46**	5,4	6,14
Test 3	7,06	4,33	**83,5**	5,13
Test 4	3,33	7,13	5,86	**83,7**

La même démarche a été faite en remplaçant l'algorithme d'apprentissage Baum-Welch par Viterbi. La matrice de confusion du maximum de vraisemblance est exprimée dans le TABLE. II.

TABLE II. MATRICE DE CONFUSION POUR LES MODÈLES (APPRENTIQQAGE VITERBI) EN POURCENTAGE (%)

	Classe 1	**Classe 2**	**Classe 3**	**Classe 4**
Test1	**85,6**	3,53	6,4	4,5
Test 2	6,06	**76,4**	6,7	10,86
Test 3	7,33	4,53	**83,86**	5,26
Test 4	4, 33	3,06	6,9	**85,73**

Nous remarquons que les résultats présentés dans la TABLE. II sont presque identiques aux résultats fournis par l'apprentissage Baum-Welch (TABLE. I).

B. Résultat pour les non modèles

Dans la partie non-modèle, nous avons d'abord construit nos corpus d'apprentissage du non-modèle comme le montre la Fig. 4. Chaque corpus d'apprentissage du non-modèle contient 350 séquences*3 c'est-à-dire, 1050 séquences d'observations d'apprentissage. Ensuite, nous avons procédé avec la même démarche pour les modèles (apprentissage par Baum-Welch dans un premier temps, puis Viterbi.

Nous avons calculé la vraisemblance de chaque séquence pour chacun des non-modèles qui ont été estimés.

Contrairement à ce que nous cherchions avec la méthode classique (celle des modèles), ici, nous cherchons la classe qui ressemble le moins à la séquence à identifier puisque nous avons appris le contraire des séquences (ce que l'on n'est pas). Pour cela, nous avons recherché le minimum de vraisemblance de chaque non-modèle. Les résultats en pourcentage (%) obtenus sur dix tests effectués sont donnés dans les matrices de confusion ci-dessous (TABLES. III et IV).

TABLE III. MATRICE DE CONFUSION POUR LES NON-MODÈLES (APPRENTIQQAGE BAUM-WELCH) EN POURCENTAGE (%)

	Non-Classe 1	**Non-Classe2**	**Non-Classe3**	**Non-Classe 4**
Test1	**81,33**	5,26	7,66	5,73
Test 2	9,8	**71,93**	8	10,26
Test 3	9,93	5,6	**76,4**	8,06
Test 4	10,93	4,8	5,4	**78,86**

TABLE IV. MATRICE DE CONFUSION POUR LES NON-MODÈLES (APPRENTIQQAGE VITERBI) EN POURCENTAGE (%)

	Non-Classe 1	**Non-Classe2**	**Non-Classe3**	**Non- Classe 4**
Test1	**81,53**	5,6	7,86	5
Test 2	8,4	**69,4**	8	14,2
Test 3	8,8	5,86	**79**	6,33
Test 4	4,7	3,86	8,53	**82,86**

Nous remarquons que les résultats présentés dans les TABLE. III et IV sont très proches. Les résultats obtenus par les non-modèles sont légèrement inférieurs à ceux obtenus par les modèles.

Ces résultats montrent la capacité de ces modèles à identifier les classes. Mais les informations contenues dans ces « non-classes » sont-elles complémentaires à celles des classes ?

V. DISCUSSION DES RÉSULTATS

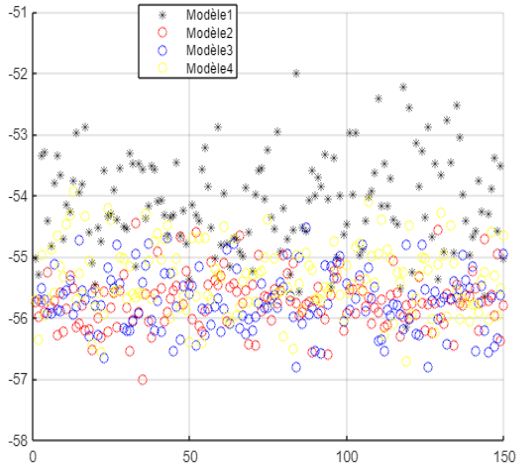
Dans cette partie, nous allons discuter des résultats obtenus par les modèles et les non-modèles.

Les (Fig. 6 (a) et (b)) montrent les valeurs de vraisemblance obtenues par les différents modèles (Fig. 6(a)) et par les non-modèles (Fig. 6 (b)) pour chaque échantillon de la classe 1 de la base de test.

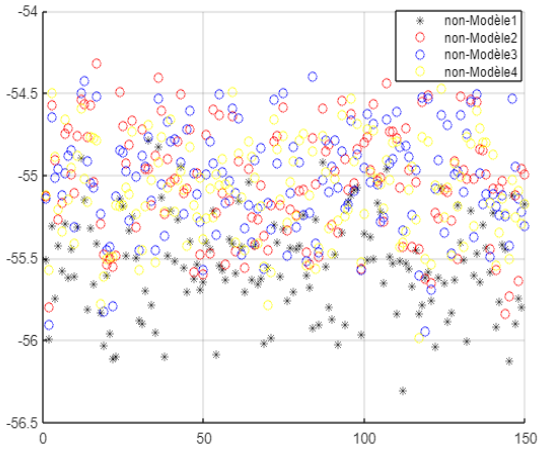
Fig. 6. Distribution des log-vraisemblances des 150 échantillons tests pour le modèle1 et le non modèle 1 (apprentissage avecViterbi)

modèle 1(a), non- modèle 1(b)

(a)



(b)



Nous voyons que globalement le modèle 1 fournit des vraisemblances supérieures aux autres modèles : capacité à reconnaitre la séquence. De même que le non-modèle 1 fournit des valeurs de vraisemblances globalement plus faibles que les autres non-modèles : capacité à non reconnaitre la séquence.

Ces résultats sont encourageants sachant qu'ils exploitent l'un et l'autre (modèles et non modèles) des corpus de données différents.

En complément de ce constat, nous avons étudié la nature des erreurs commises par les modèles et les non-modèles.

Dans le tableau ci-dessous (TABLE. V), nous montrons un petit aperçu sur les erreurs (échantillons qui font partie de la classe 1 et non-classe 1 mais, mal identifiés par les modèles ou non-modèles.

TABLE. V. Aperçu sur quelques échantillons pour le modèle 1 et le non-modèle 1

Numéros d'échantillons				
Classe1=1				
Classe1=2	6		**114**	**146**
Classe1=3				
Classe1=4		**16**		
Non-classe1=1				**146**
Non-classe1=2	6			
Non-classe1=3		**16**		
Non-classe1=4			**114**	

D'après ce tableau, nous voyons différentes situations possibles :

- Echantillon 6 : même erreur pour modèle et non-modèle.
- Echantillons 16 et 14 : erreurs différentes pour modèle et non-modèle.
- Echantillon 146 : erreur pour modèle mais bonne identification de non-modèle.

VI. CONCLUSION

Dans les travaux présentés, nous avons utilisé les Modèles de Markov Cachés (MMCs) afin d'évaluer le niveau de dégradation d'un système industriel. Des données synthétiques ont été produites en considérant un système sujet à trois modes de défaillance et d'un mode de fonctionnement normal.

A l'aide des résultats des différents essais effectués, nous avons montré la capacité des non-modèles utilisant les échantillons des autres classes (non-classes) à identifier correctement les échantillons de la classe considérée avec des taux d'identification similaires à ceux obtenus par les modèles.

La suite de nos travaux va consister à développer un outil algorithmique permettant d'exploiter les informations contenues dans les non-classes pour améliorer l'apprentissage des modèles.

REFERENCES

[1] H.-Y. Hsu, G. Srivastava, H.-T. Wu, et M.-Y. Chen, « Remaining useful life prediction based on state assessment using edge computing on deep learning », Computer Communications, vol. 160, p. 91-100, 2020.

[2] G.Cheng, X.inzhiWang et Y.He « Remaining useful life and state of health prediction for lithium batteriesbased on empirical mode decomposition and a long and short memory neural network »,Energie, vol. 232, Article 121022, 1 October 2021

[3] G.Qiu, Y.Gu et J. Chen, « Selective health indicator for bearings ensemble remaining useful life prediction with genetic algorithm and Weibull proportional hazards model », Measurement, vol. 150, Article 107097,janvier 2020

[4] X.inLi, Y.Ma, J.Zhu, « An online dual filters RUL prediction method of lithium-ion battery based on unscented particle filter and least

squares support vector machine », Measurement, vol. 184, Article 109935,novembre 2021

[5] M.SubKim, D.Kim, S.Lee, « Face recognition using the embedded HMM with second-order block-specific observations », Pattern Recognition,vol.36, p.2723-2735,november 2003

[6] J.-M. Liu, M. You, Z. Wang, G.-Z. Li, X. Xu, et Z. Qiu, « Cough event classification by pretrained deep neural network », BMC MEDICAL INFORMATICS AND DECISION MAKING, vol. 15, nº 4, Art. nº 4, novembre. 2015

[7] C. Bunks, D. McCarthy et T. Al-Ani, « Condition-based maintenance of machines using hidden Markov models», Mechanical Systems and Signal Processing, vol.14, p597-612, juillet 2000

[8] H. M. Ertunc, K. A. Loparo et H. Ocak, «Tool wear condition monitoring in drilling operations using hidden Markov models (HMMs)».International Journal of Machine Tools and Manufacture,vol.41, p1363-1384, 2001

[9] J. M. Lee, S. J. Kim, Y. Hwang et C. S. Song, «Diagnosis of mechanical fault signals using continuous hidden Markov model». Journal of Sound and Vibration, vol.276, p 1065- 1080, 2004

[10] R. B. Chinnam et P. Baruah, «Autonomous diagnostics and prognostics through competitive learning driven HMM-based clustering» .Neural Networks, Proceedings of the International Joint Conference on. Vol. 4. IEEE, 2003.

[11] R. B. Chinnam et P. Baruah, « HMMs for diagnostics and prognostics in machining processes». International Journal of Production Research,vol.43, p1275-1293, 2005.

[12] X. Zhang, R. Xu, C. Kwan, S. Y. Liang, Q. Xie et L. Haynes, « An integrated approach to bearing fault diagnostics and prognostics». American Control Conference IEEE, 2005.

[13] A. Giantomassi, F. Ferracuti, A. Benini, G. Ippoliti, S. Longhi, et A. Petrucci, « Hidden Markov Model for Health Estimation and Prognosis of Turbofan Engines». ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. American Society of Mechanical Engineers, 2011.

[14] D. A. Tobon-Mejia, K. Medjaher, N. Zerhouni, et G. Tripot, « Hidden Markov models for failure diagnostic and prognostic». Prognostics and System Health Management Conference (PHM-Shenzhen), 2011. IEEE, 2011.

[15] D. A. Tobon-Mejia, K. Medjaher, N. Zerhouni, et G. Tripot , « A data-driven failure prognostics method based on mixture of Gaussians hidden Markov models». Reliability, IEEE, vol.61, p491-503,2012.

[16] T.Aggab,P.Vrignat, M.Avila, F.kratz, « Estimation du niveau de dégradation par un modèle de Markov caché multi-flux»,QUALITA' 2015,Nancy,France, 2015.

[17] L.Cao, S. Yu, V. Kumar, « Nonoccurring Behavior Analytics: A New Area», IEEE Intelligent System,vol.30, p.4-11 November 2015.

[18] L. R. Rabiner, « A tutorial on hidden Markov models and selected applications in speech recognition», Proceedings of the IEEE,vol.77,p 257 – 286, 1989.

[19] S. Aupetit, N. Monmarch,S.Mhamed, « Utilisation des modèles de Markov Cachés pour la reconnaissance robuste d'images : apprentissage par colonie de fourmis, algorithme génétique et essaim particulaire», Siarry P., Ed., Optimisation en traitement du signal et de l'image, Traitement dusignal et de l'image, Traité IC2, p. 245–269, Hermès Science, Paris, Février 2007

[20] B.Robles, « Etude de la pertinence des paramètres stochastiques sur des modèles de Markov cachés », thèse de doctorat, université d'Orléans, 2013.