



FONDATION D'ENTREPRISE

FEREC

Correction des températures - CorrelTemp

Fondation d'entreprise FEREC

Appel à projets 2020 : Intelligence artificielle (IA) appliquée aux infrastructures en service



SYSTRA

Auteurs :

LEO CAZENILLE, BAPTISTE BONY, STEPHANE SLUSAREK
SIXENSE

Sommaire

Synthèse	2
1. Introduction.....	3
1.1 Méthode de correction de l'effet de la température	4
1.2 Résultats	10
2. Validation par simulation de dégradations.....	15
3. Visualisation des degrés de dégradation	17
3.1 Méthode	17
3.2 Résultats	18
3.3 Discussions et conclusion	20
4. Références.....	21

	<i>Ind</i>	<i>Date</i>	<i>Rédaction</i>	<i>Vérification</i>	<i>Contrôle qualité</i>
Révisions	0	16/10/2021	L. Cazenille	S. Slusarek	
	A	18/10/2021	L. Cazenille	A. Basth	
	B	19/10/2021	L. Cazenille	F. Boureau	
	C	27/10/2021	L. Cazenille	S. Slusarek	
	D	29/10/2021	L. Cazenille	M. Beth	
	E	19/11/2021	S. Slusarek		

Synthèse

Dans le domaine de la construction, le suivi et diagnostic de bâtiments au long de leur cycle de vie peut être nécessaire pour détecter l'apparition de dégradations (e.g. déformations, déplacements, fissures, etc.). Ce suivi est typiquement établi à partir de données capturées en temps réel par des capteurs positionnés sur le bâtiment d'étude, ou dans leur entourage. Cependant, ces séries temporelles sont souvent biaisées par leur environnement -- en particulier par la température, qui a des effets à la fois sur la capture des données, mais aussi directement sur les bâtiments (effets saisonniers). Dans cette étude, nous développerons une méthodologie permettant d'extraire ces biais environnementaux des séries temporelles des capteurs pour obtenir des données non-biaisées. Ces données corrigées permettront de mieux identifier la présence de dégradation. Notre méthodologie sera basée sur des méthodes d'apprentissage automatique (« random forests », réseaux de neurones récurrents, etc.), et s'inspirera de l'état de l'art de la suppression de biais dans des séries temporelles.

1. Introduction

Les structures et infrastructures subissent régulièrement des effets de l'environnement, ce qui entraîne inévitablement des détériorations (déformations, déplacements, fissures, etc.). Dans un souci de sûreté publique et pour éviter les interruptions de services, il est donc d'une importance critique de pouvoir détecter ces détériorations le plus tôt possible, et de suivre leur progression. D'autre part, il a été prouvé qu'une surveillance continue d'une infrastructure permet de réduire considérablement ses coûts d'opération et de maintenance [15].

Le processus de surveillance de la santé des structures et ouvrages est donc généralement établi de façon continue (potentiellement pendant tout le cycle de vie du système surveillé) à partir de données d'observations caractérisant différents types de changements ou de dégradations (déplacements, fissures, etc.). Ces données sont obtenues en temps réel à partir de capteurs placés sur la structure, ou à ses alentours (théodolites, fissuromètres, tilt mètres, etc.).

Cependant, identifier la présence de dégradations à partir de ces données est un processus potentiellement complexe, qui fait appel à des méthodes d'analyse statistiques pour séparer les différentes sources de variation des phénomènes observés [2,6].

En effet, les dégradations peuvent être causées par différents types d'effets environnementaux (érosion, fatigue des bâtiments, catastrophes naturelles, etc.).

En particulier, un grand nombre de structures et d'ouvrages sont impactés par la température de l'air ambiant [14]. Cette température change de façon saisonnière, mais aussi de façon journalière (cycles jours-nuits). Elle peut avoir un effet direct sur les bâtiments, notamment des effets de déplacements ou de changement d'orientation. Cependant, la plupart de ces effets sont réguliers au cours de l'année, et ne correspondent pas à une dégradation réelle, et peuvent donc être vus comme un biais sur les mesures. Il peut donc s'avérer nécessaire de retirer l'influence de la température sur ces mesures, pour pouvoir identifier si elles comportent bien des dégradations.

Dans cette étude, nous décrivons une méthode qui permet d'identifier l'influence systémique de la température sur les mesures (signaux) d'auscultation de plusieurs ouvrages et structures. Cette influence sera considérée comme un biais à retirer des signaux d'entrée, pour obtenir des signaux corrigés, qui pourront ensuite être utilisés pour détecter la présence de dégradation.

Cette méthodologie repose en grande partie sur l'obtention d'un modèle liant l'évolution de la température au cours du temps avec les signaux des mesures étudiées. Nous testerons donc plusieurs méthodes pour modéliser l'effet de la température, que ce soit des modèles inspirés de l'état de l'art pour l'auscultation (**HST**) ou des modèles par apprentissage automatique (« Machine Learning »).

Ensuite, nous validerons nos résultats en les testant sur des cas présentant des dégradations tirées de mesures réelles de capteurs de différents types (extensomètres, théodolites, etc.) positionnés sur différentes structures. Nous étendrons notre processus de validation aussi à des jeux de données modifiés pour comporter des dégradations artificiellement ajoutées aux jeux d'origine.

Finalement, nous présenterons une méthode permettant de détecter la présence de dégradations en prenant compte de l'incertitude sur les mesures.

1.1 Méthode de correction de l'effet de la température

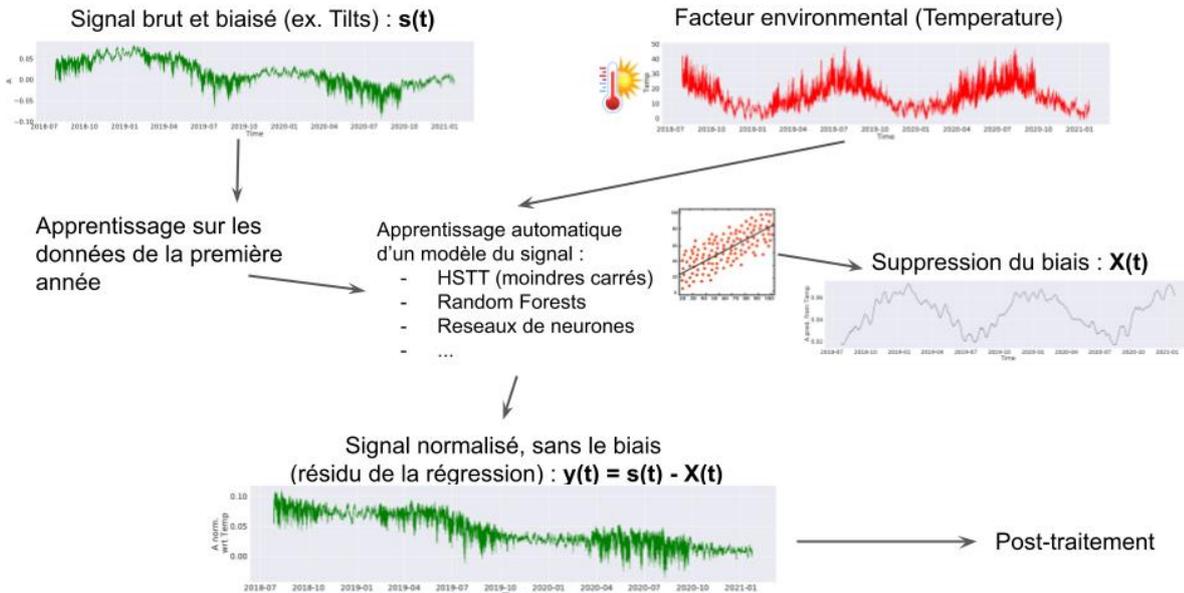


Figure 1. : Organisation de la méthodologie de correction de l'effet de la température

Notre méthodologie a pour objectif d'identifier et d'extraire l'effet de la température sur les mesures d'auscultation. Un diagramme de son organisation est présenté en Figure 1.

La méthodologie utilise deux signaux d'entrée : un signal « brut » $S(t)$ correspondant à la mesure d'auscultation, biaisé par la température, et un signal correspondant à la température de l'air observée proche du capteur d'auscultation.

Ces signaux sont tout d'abord segmentés par l'utilisateur en deux périodes de temps successives : une partie dite de "référence", que l'on assumera non - ou faiblement - biaisée (par exemple la première année des données), et une partie "biaisée" (par exemple les données des années suivantes).

En utilisant les données de la période de "référence", nous entraînons un modèle représentant le signal brut à partir du signal de température. Cela permet de modéliser l'effet direct de la température sur le signal brut et de l'extraire sous la forme d'une fonction de biais $X(t)$. A condition que ce modèle soit suffisamment réaliste, il est donc possible de calculer un signal corrigé, et donc sans l'influence de la température par la formule :

$$y(t) = s(t) - X(t)$$

Ce signal corrigé peut ensuite subir d'autres traitements additionnels, comme un recentrage (pour faciliter la comparaison avec le signal "brut"), un filtrage (pour ne voir que les effets sur le long terme), etc.

La principale difficulté rencontrée dans l'application de cette méthodologie est donc de trouver un bon modèle $M(t, T)$. Dans les sections suivantes, nous testerons et comparerons donc des modèles de plusieurs familles différentes. La Table 1 liste les différents modèles étudiés.

Nous évaluerons les résultats de ces modèles à partir de leur coefficient de détermination (couramment appelé « le score de R^2 ») qui est une valeur entre 0 et 1 mesurant la qualité des prédictions du modèle par rapport aux signaux bruts -- un score plus élevé correspondant à des prédictions plus précises.

Méthode	Type	Nombre de paramètres	Variables d'entrées	Normalisation des variables d'entrées
HSTT	Analytique	10	2	Aucune
HSTT4	Analytique	14	2	Aucune
MLP-HST	Réseaux de neurones	39	4	Min-max
RF	Forêt aléatoire	-	10	Aucune
LSTM	Réseaux de neurones	2392	2*100	Min-max

Table 1. : Liste des méthodes utilisées

1.1.1 Modélisation du signal par HST et HSTT

Le modèle HST [6, 17] permet d'interpréter les données d'auscultation mesurées, pour la surveillance d'un ouvrage. Il a été développé principalement pour comprendre le comportement des barrages sur le long terme. Cette méthode est basée sur une analyse statistique appelée Hydrostatique-Saison-Temps (HST) et permet de séparer les différentes sources de variation du phénomène étudié.

Le modèle HST considère que les mesures brutes peuvent être modélisées par une superposition de trois états :

- Effet hydrostatique (H) : variation réversible du niveau de la retenue.
- Effet saisonnier (S) : influence réversible saisonnière (température).
- Effet temps (T) : évolution irréversible du phénomène au cours du temps

On a l'expression de base du modèle HST :

$$Y_{\text{hst}}(t) = Y_0 + H(t) + S(t) + T(t) + \varepsilon$$

Avec :

$$H(t) = a_1 z(t) + a_2 z^2(t) + a_3 z^3(t) + a_4 z^4(t)$$

$$S(t) = b_1 \sin(\theta) + b_2 \cos(\theta) + b_3 \sin^2(\theta) + b_4 \sin(\theta)\cos(\theta)$$

$$T(t) = c_1 \tau(t) + c_2 e^{\tau(t)} + c_3 e^{-\tau(t)}$$

En considérant :

$$z(t) = \frac{Z(t) - \mu(Z)}{\sigma(Z)}$$

$$\theta(t) = \frac{2\pi t}{365.5}$$

$$\tau(t) = \frac{t - t_i}{t_f - t_i}$$

Y_0 est un paramètre. Nous le calibrons en utilisant la méthode du Simplex (Nelder-mead) [16] pour obtenir le modèle avec la plus grande capacité prédictive (en utilisant le coefficient de détermination comme fonction objectif).

Le résidu $e(t)$ est défini par l'écart entre la mesure $y(t)$ et la valeur calculée $Y_{hst}(t)$. Il intègre les erreurs expérimentales, l'imprécision du modèle, et surtout, le biais environnemental (en particulier de la température) :

$$e(t) = y(t) - Y_{hst}(t)$$

On utilisera comme méthode d'ajustement une régression linéaire multiple. Ces paramètres seront estimés par moindres carrés.

Le modèle HST a été développé pour les barrages et il ne prend pas en compte la température ambiante. Pour élargir son champ d'application, des modèles **HSTT**, avec le second T pour la température, existent. Ils sont peu utilisés. Nous en testerons 2 variantes, nommés ici HSTT et HSTT4 par souci de simplification.

Les modèles **HST et HSTT** présentent plusieurs avantages : ils sont interprétables analytiquement, faciles à calibrer (avec peu de paramètres), et ont un faible coût de calcul. Cependant, ils ont aussi un certain nombre de limitations : en particulier, ces modèles ne capturent pas toutes les relations entre les variables explicatives (seulement celles décrites dans les équations).

1.1.2 MLP-HST : Réseaux de neurones utilisant les variables d'HSTT

Les modèles basés sur HST se basent sur des descriptions analytiques (équations) pour exprimer le lien entre signal d'auscultation et variables explicatives -- ce qui a l'avantage d'avoir un modèle simple à interpréter. Cependant, ce type de modèle ne permet pas de représenter toutes les façons dont les variables interagissent pour aboutir au résultat final, et donc ne capture pas toutes les dynamiques du système. Une alternative est d'utiliser d'autres types de modèles avec des capacités de représentation plus élevées. Notamment, les réseaux de neurones artificiels [19], un type de modèle très utilisé en Intelligence Artificielle, sont considérés comme des approximateurs universels de fonctions. Cette haute capacité prédictive se fait au prix d'une faible interprétabilité -- on parle donc de modèle "black box". Les réseaux neuronaux sont bio-inspirés, et correspondent à des graphes d'unités de calculs basées sur des formulations simplifiées des neurones biologiques. Ces "neurones" sont interconnectés entre eux par différentes topologies de réseaux, suivant le type de modèle.

Le modèle le plus classique suit une topologie en couches (« feed-forward layers ») contenant chacune une collection de neurones prenant en entrée les sorties de la couche précédente et connectés aux entrées des neurones de la couche suivante. Ce type de réseau est appelé « Perceptron Multi-Couches » (**MLP** : « Multi-Layer Perceptron »).

L'article de Simon et al. [6] propose d'utiliser un **MLP** à la place du modèle **HST**, et montre que ce premier permet d'obtenir une plus grande capacité prédictive comparé à **HST**. Nous proposons de tester ce modèle dans notre étude. Comme ce modèle prend en entrée des variables similaires à celles utilisées par HST, nous l'appellerons « **MLP-HST** ».

Nous utilisons une topologie similaire à celle décrite dans [6] avec 4 neurones de couche cachée et 1 neurone de sortie. Elle est illustrée dans la Figure 2.

Pour chaque pas de temps t , le modèle prend 4 variables explicatives en entrée : le temps (t), la saison ($\cos(S)$ et $\sin(S)$), et la température.

La sortie correspond au signal prédit au pas de temps t .

Nous utilisons la fonction de tangente hyperbolique comme fonction d'activation des neurones (« tanh »). L'apprentissage du MLP se fait en utilisant l'algorithme ADAM.

Notre implémentation en Python utilise les bibliothèques PyTorch [8] et sklearn [9]. Les hyperparamètres utilisés dans ce modèle sont regroupés en Table 2.

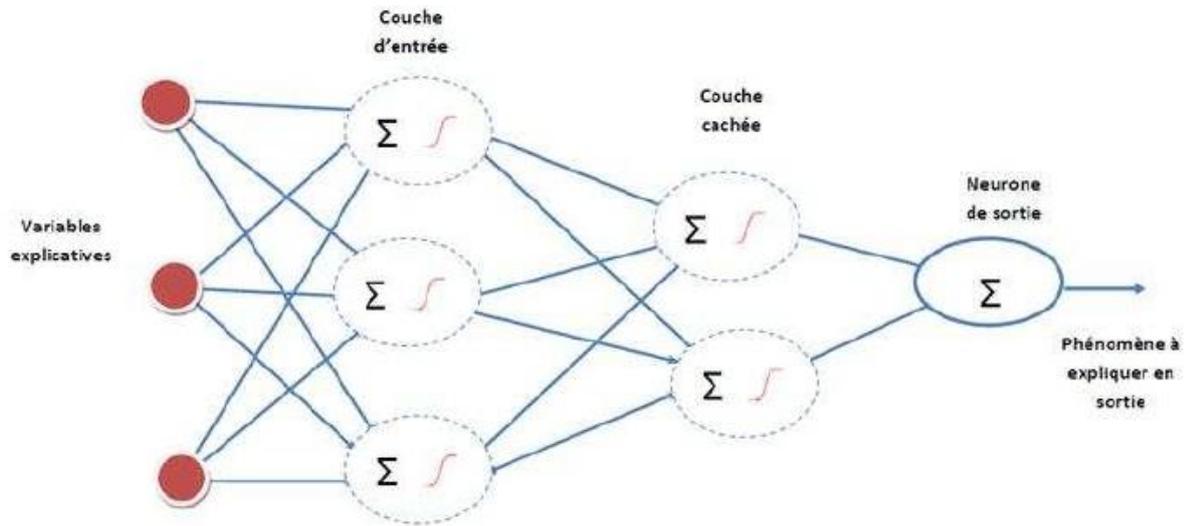


Figure 2. Architecture du réseau de neurones utilisé dans le modèle HST-MLP (figure tirée de [7])

Methode	Paramètre	Valeur / Domaine
HST-MLP [7]	Nombre total de paramètres	39
	Paramètres d'entrées	4
	Topologie	4 neurones couche cachée, 1 sortie
	Algorithme d'apprentissage	ADAM [8]
	Initialisation des poids	Xavier uniforme
	Nombre d'epochs	400
	Learning rate	0.01
	Batch size	512
	Random Forest [12]	Nombre total de paramètres
Paramètres d'entrées		10 (cf section RF)
Nombre d'arbres de décision		100
Splits de cross-validation		5 KFold
Critère de split		Erreur au carré
Profondeur		1, 2, 3, 5, 7, 10, 20, inf
Complexity parameter used for Minimal Cost-Complexity Pruning		0.0, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35
min_samples_split		2, 3, 4, 5, 6, 7
LSTM [11]	Nombre de features après l'élimination récursive (RFE)	2, 3, toutes
	Nombre total de paramètres	2392
	Paramètres d'entrées	2*100
	Topologie	2 couches LSTM (5 unités) + 1 couche dense (5 neurones) + 1 sortie
	Algorithme d'apprentissage	ADAM [8]
	Initialisation des poids	Xavier uniforme
	Nombre d'epochs	400
	Learning rate	0.01
Batch size	512	

Table 2. : Hyperparamètres utilisés dans les méthodes d'apprentissage automatique

1.1.3 RF : Méthode basée sur des Forêts aléatoires

Nous proposons d'utiliser des « forêts aléatoires » (« Random forests » en anglais : « **RF** ») comme modèles pour représenter le signal. Il s'agit d'un autre modèle par apprentissage automatique, qui s'inspire d'un autre modèle prédictif : les arbres de décisions.

Ces derniers lient de façon explicite les valeurs des variables des variables cibles et les variables d'entrée par recombinaisons à l'intérieur d'une structure d'arbres (chemins). Ils peuvent être utilisés pour des tâches de régression (ce qui correspond au cas où les variables d'entrée et les variables cibles sont des valeurs continues). Pour pouvoir les calibrer pour un problème donné, les arbres de décision peuvent être entraînés par apprentissage supervisé : on utilise un ensemble de données pour lesquelles on connaît les valeurs des variables cibles pour construire l'arbre, puis on extrapole les résultats à l'ensemble des données de test. Les arbres de décision font partie des algorithmes les plus populaires en apprentissage automatique.

Les forêts aléatoires regroupent les résultats d'une collection d'arbres de décisions, chacun entraînés sur un sous-ensemble différent des données. Ensuite, les prédictions des forêts aléatoires sont obtenues par vote majoritaire des arbres de décision (« ensemble learning »). Cette approche par ensemble permet d'obtenir

des capacités de représentation et de prédiction (par rapport au coefficient de détermination) généralement supérieures à celles obtenues par de simples arbres de décisions uniques.

Nous utilisons l'implémentation des forêts aléatoires de la librairie Python « scikit-learn » [9]. Elle se base sur la méthode d'apprentissage d'arbres « perturb-and-combine » [16].

Les hyperparamètres de cette méthode, et leur valeurs (ou domaines) sont listés dans la Table 2.

La méthode des forêts aléatoires a beaucoup d'hyperparamètres qui peuvent influencer sur leur résultats (par exemple le coefficient de détermination). Il est donc courant de tester plusieurs valeurs pour certains hyperparamètres (par exemple la profondeur maximale des arbres), et de ne garder que les meilleurs résultats pour chaque jeu de données. Pour ce faire, nous utilisons un processus nommé « Grid Search » dans lequel tous les sets d'hyperparamètres d'intérêt sont testés.

La méthode RF utilise des variables d'entrées différentes (et plus complètes) de celles utilisées pour le modèle HST. En effet, le modèle RF n'a pas vocation à être un modèle facilement interprétable (modèles en « boîtes blanches » / « white box models »), mais plutôt un modèle qui permet de capturer le signal de la façon la plus précise possible (modèles en « boîtes noires » / « black box models »). Par conséquent, nous lui fournissons aussi des variables d'entrées qui ne peuvent pas facilement être interprétables, mais qui contiennent plus d'informations que celles utilisées pour le modèle HST.

Il peut être judicieux (du point de vue des scores de coefficients de détermination des résultats) de ne prendre en compte qu'un sous-ensemble de variables d'entrées par le modèle RF, suivant le jeu de données. Pour ce faire, nous utilisons la méthode « RFE » (« Recursive Feature Elimination ») pour sélectionner soit 2, soit 3, soit toutes les variables d'entrées - comme pour la sélection des hyperparamètres, seuls les meilleurs résultats par jeu de données sont conservés.

1.1.4 LSTM : Méthode basée sur des réseaux de neurones récurrents

Nous proposons une autre méthode par apprentissage automatique, et faisant appel à un modèle de réseaux de neurones. En effet, un modèle classique et d'expressivité élevée pour représenter des signaux temporels complexes est d'utiliser des « réseaux de neurones récurrents ». Ce sont des réseaux de neurones qui comportent des connexions récurrentes - contrairement aux réseaux de neurones « feed-forward », comme les MLP, qui n'en comportent pas. Ces connexions récurrentes permettent de capturer des états internes qui permettent de mieux représenter l'évolution de phénomènes temporels de pas de temps en pas de temps.

Cependant, ces réseaux sont généralement plus difficiles à entraîner que de simples réseaux feed-forward (problème de « disparition du gradient » en utilisant un apprentissage par « rétropropagation du gradient »). Des architectures particulières permettent d'adresser partiellement ce problème -- en particulier les réseaux de type « Long Short-Term Memory » (ou **LSTM**) [10] qui contiennent des mécanismes pour propager les erreurs aux pas d'apprentissage antérieurs, et éviter la plupart des problèmes de disparition du gradient. Ce type de réseau est constitué d'unités computationnelle modélisant des états interne de court et long termes -- une de ces unités est illustré dans la Figure suivante :

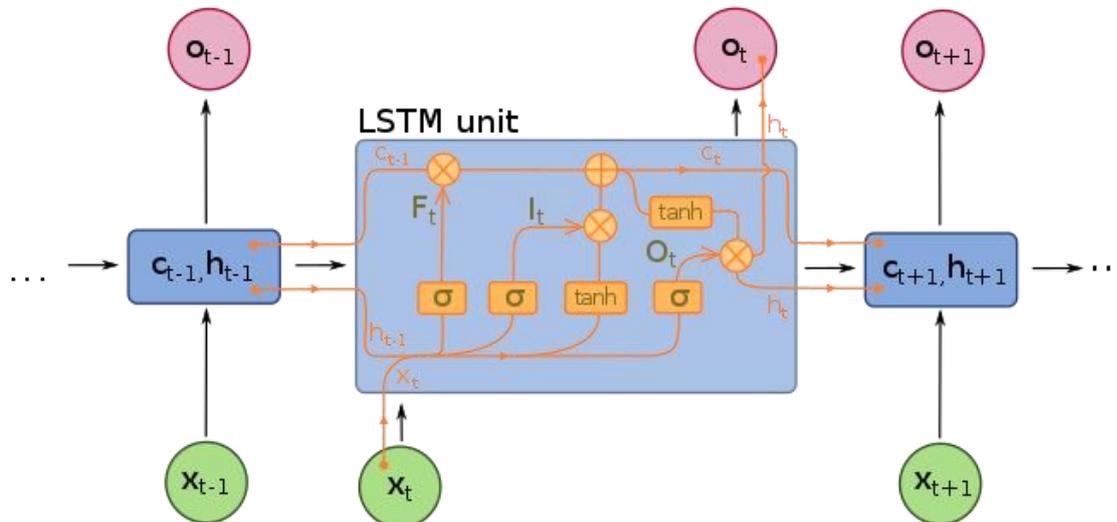


Figure représentant une seule unité d'un réseau LSTM. Image tirée de https://fr.wikipedia.org/wiki/R%C3%A9seau_de_neurones_r%C3%A9currents#Long_short-term_memory

Dans notre étude, nous proposons d'utiliser ce type de modèle pour représenter le signal d'auscultation étudié, et le prédire à partir de la courbe de température.

Pour prédire la valeur du signal au temps $t + 1$, le modèle **LSTM** utilise en entrée les deux vecteurs de valeurs suivants :

- Un vecteur correspondant au temps $t - 100 \dots t$ des 100 derniers pas de temps avant $t + 1$.
- Un vecteur correspondant aux valeurs de la température à tous les 100 derniers pas de temps avant $t + 1$.

Nous utilisons un réseau avec les couches suivantes :

- 1) Une couche de 5 unités LSTM
- 2) Une couche de 5 unités LSTM
- 3) Une couche de MLP (recombinaison linéaire) de 5 neurones
- 4) Une sortie, correspondant au signal au temps $t + 1$

Notre implémentation en Python utilise les bibliothèques PyTorch [9] et sklearn [10]. L'apprentissage se fait en utilisant l'algorithme ADAM et les hyper-paramètres listés dans la Table 2.

1.2 Résultats

Nous testons les modèles présentés précédemment d'abord au niveau de leur capacité prédictive sur une panoplie variée de jeux de données d'auscultation, et ensuite sur leur capacité à servir de base pour corriger les effets dus à la température.

Les jeux de données considérés comportent des données issues de théodolites ("Cyclops"), des fissuromètres, tiltmètres, et extensomètres, placés sur différents types de structures ou ouvrages (ponts, immeubles, barrages, gares, etc.) à différentes périodes de leur cycle de vie (pendant le chantier, ou après construction).

Les méthodes **HST-MLP**, **RF** et **LSTM** sont stochastiques (c'est-à-dire qu'elles donnent des résultats différents à chaque fois qu'elles sont entraînées sur le même jeu de données), alors que les méthodes **HST** et **HSTT** sont déterministes. Pour étudier les résultats des méthodes stochastiques, nous considérons des résultats moyennés sur 20 instances (apprentissage et application) différentes de chaque méthode pour chaque jeu de données.

Dans un premier temps, nous nous concentrerons sur les résultats des modèles analytiques basés sur **HST**, et comparerons leur capacité prédictive, par rapport aux coefficients de détermination (R^2) obtenus sur différents jeux de données. Ces résultats sont listés dans la Figure 3. Nous séparons les résultats des

modèles **HST** et **HSTT** de ceux obtenus uniquement par leur composantes $H(t)$ (« **Polynomial** »), $S(t)$ (« **Sinusoïdal** ») et $T(t)$ (« **Exponentiel** »).

Comme les modèles **HST** et **HSTT** incluent toutes ces composantes, ils ont généralement des coefficients de détermination plus élevés que ces composantes prises individuellement.

Nos résultats indiquent que le modèle **HSTT** est généralement celui avec les coefficients de détermination les plus élevés (dans 22/39 cas), suivi de **HST** (dans 3/39 cas) avec des résultats souvent très proches de ceux d'**HSTT**. Certains jeux de données peuvent être plus facilement représentés juste par un modèle polynomial ne prenant en compte que la température - c'est le cas lorsque la température est très corrélée au signal (sans bruit/biais additionnel sur la mesure). D'autres jeux de données ont les meilleurs résultats en n'utilisant uniquement la composante exponentielle (en particulier quand la période de temps du jeu de donnée est relativement courte).

Il est à noter que les modèles **HST** et **HSTT** sont calibrés en utilisant une méthode par moindre carrée. Cette méthode d'optimisation ne garantit pas de trouver les meilleures valeurs de paramètres possibles (seulement un « optimal local »), en particulier quand le nombre de paramètres devient plus grands (c'est le cas de **HSTT**) -- il serait donc possible d'améliorer encore ces résultats en utilisant d'autres méthodes de calibrations (méthodes quasi-newton, méta-heuristiques, etc.).

	POLYNOMIAL	SINUSOIDAL	EXPONENTIEL	THERMIQUE	THERMIQUE 4
CYCLOPS 01	0.812078	0.595802	0.519553	0.820236	0.859765
CYCLOPS 02	0.829445	0.759224	0.723058	0.880766	0.892128
CYCLOPS 03	0.772146	0.725560	0.684324	0.841493	0.845438
CYCLOPS 04	0.704323	0.825921	0.801353	0.769209	0.763886
CYCLOPS 05	0.324861	0.435911	0.431426	0.320521	0.329282
CYCLOPS 06	0.641078	0.646340	0.619502	0.651907	0.647769
CYCLOPS 07	0.352048	0.183408	0.136624	0.304102	0.360511
CYCLOPS 08	0.772988	0.691784	0.663190	0.785139	0.798046
CYCLOPS 09	0.784083	0.642700	0.580205	0.815657	0.837460
CYCLOPS 10	0.778619	0.738539	0.692271	0.842465	0.861228
CYCLOPS 11	0.773486	0.769420	0.714278	0.838728	0.859439
CYCLOPS 12	0.674997	0.693530	0.634185	0.711212	0.741788
CYCLOPS 13	0.505786	0.284944	0.246978	0.500129	0.513976
CYCLOPS 14	0.822233	0.608018	0.580554	0.846202	0.845254
CYCLOPS 15	0.842980	0.602912	0.576982	0.871195	0.871413
CYCLOPS 16	0.849336	0.606275	0.550968	0.866798	0.881154
CYCLOPS 17	0.857305	0.678919	0.604143	0.869230	0.895601
CYCLOPS 18	0.821435	0.653611	0.576239	0.828636	0.871041
FISSUROMÈTRE 01	0.846005	0.587959	0.582683	0.845690	0.964754
FISSUROMÈTRE 02	0.995140	0.580470	0.567961	0.888253	0.905613
FISSUROMÈTRE 03	0.930354	0.594777	0.581515	0.935164	0.947000
FISSUROMÈTRE 04	0.817814	0.761701	0.757609	0.936620	0.958545
TILTMÈTRE 01	0.399637	0.287586	0.268969	0.181649	0.190893
TILTMÈTRE 02	0.898476	0.626879	0.629020	0.776818	0.803002
TILTMÈTRE 03	0.971588	0.729147	0.145596	0.914614	0.918139
TILTMÈTRE 04	0.046657	0.095494	0.768223	0.766686	0.786675
TILTMÈTRE 05	0.093322	0.108384	0.926899	0.841219	0.859049
PONT 1 (EXTENSOMÈTRE)	0.935454	0.860634	0.048235	0.972905	0.974365

	POLYNOMIAL	SINUSOIDAL	EXPONENTIEL	THERMIQUE	THERMIQUE 4
PONT 2 (EXTENSOMÈTRE)	0.925937	0.828693	0.096663	0.945762	0.948557
PONT 3 (EXTENSOMÈTRE)	0.969487	0.899821	0.081744	0.965413	0.968975
CYCLOPS P51 01	0.546600	0.597657	0.584225	0.870566	0.878978
CYCLOPS P51 02	0.575577	0.648929	0.638463	0.918360	0.929375
CYCLOPS P51 03	0.555197	0.655141	0.646467	0.910290	0.915559
BARRAGE 01 (PENDULE)	0.018116	0.014337	0.977353	0.017501	0.015447
BARRAGE 02 (PENDULE)	0.013113	0.029274	0.958275	0.015830	0.005969
BARRAGE 03 (PENDULE)	0.017241	0.019487	0.970589	0.016197	0.004262
CYCLOPS 120X	0.049861	0.066792	0.885338	0.053591	0.054826
CYCLOPS 120Y	0.049861	0.066792	0.885338	0.053591	0.054826
CYCLOPS 120Z	0.033061	0.104101	0.641657	0.445406	0.486691

Figure 3. : Résultats des scores R^2 des méthodes HSTT et HSTT4, et pour chacune de leurs composantes

Dans un second temps, nous comparerons les résultats obtenus par la méthode **HSTT4** et celles obtenues par les autres méthodes étudiées (**HST-MLP**, **RF**, **LSTM**).

	THERMIQUE 4	HST-MLP	RANDOM FOREST	LSTM
CYCLOPS 01	0.856940	0.793837	0.850958	0.842931
CYCLOPS 02	0.889628	0.826525	0.767964	0.746789
CYCLOPS 03	0.843048	0.779946	0.804133	0.786870
CYCLOPS 04	0.761391	0.698288	0.798364	0.778454
CYCLOPS 05	0.325722	0.262619	0.573999	0.536289
CYCLOPS 06	0.645194	0.582091	0.760171	0.735312
CYCLOPS 07	0.358282	0.295180	0.408168	0.403953
CYCLOPS 08	0.795123	0.732020	0.788852	0.767326
CYCLOPS 09	0.834605	0.771502	0.841728	0.835136
CYCLOPS 10	0.858684	0.795582	0.816603	0.805890
CYCLOPS 11	0.856870	0.793767	0.815591	0.800851
CYCLOPS 12	0.737755	0.674652	0.771641	0.749573
CYCLOPS 13	0.510919	0.447817	0.683329	0.658928
CYCLOPS 14	0.843146	0.780044	0.900697	0.913733
CYCLOPS 15	0.869084	0.805981	0.907040	0.924438
CYCLOPS 16	0.879103	0.816001	0.904971	0.919035
CYCLOPS 17	0.893173	0.830070	0.894216	0.908448
CYCLOPS 18	0.868719	0.805617	0.871799	0.870807
FISSUROMÈTRE 01	0.964752	0.905318	0.945869	0.926749
FISSUROMÈTRE 02	0.904917	0.850623	0.972248	0.977317
FISSUROMÈTRE 03	0.946527	0.888647	0.963499	0.975104
FISSUROMÈTRE 04	0.958458	0.899745	0.928769	0.896798
TILTÈMÈTRE 01	0.192092	0.154076	0.872623	0.842203
TILTÈMÈTRE 02	0.802091	0.764075	0.948488	0.984460
TILTÈMÈTRE 03	0.917941	0.859057	0.965516	0.975431
TILTÈMÈTRE 04	0.786516	0.743643	0.341049	0.342174
TILTÈMÈTRE 05	0.859033	0.816160	0.606089	0.614003
PONT 1 (EXTENSOMÈTRE)	0.974390	0.893531	0.945878	0.971289
PONT 2 (EXTENSOMÈTRE)	0.948549	0.868985	0.938334	0.974512
PONT 3 (EXTENSOMÈTRE)	0.969002	0.924759	0.953654	0.971890
CYCLOPS P51 01	0.878184	0.822607	0.802700	0.766894
CYCLOPS P51 02	0.929477	0.871744	0.891925	0.849320
CYCLOPS P51 03	0.915644	0.860067	0.873310	0.837492
BARRAGE 01 (PENDULE)	0.015887	0.060620	0.646867	0.631651
BARRAGE 02 (PENDULE)	0.006355	0.070172	0.616138	0.594571
BARRAGE 03 (PENDULE)	0.004665	0.067267	0.632035	0.616955
CYCLOPS 120X	0.054875	0.032769	0.638004	0.588398
CYCLOPS 120Y	0.054875	0.032769	0.638004	0.588398
CYCLOPS 120Z	0.485472	0.443570	0.692300	0.667375

Figure 4 : Comparaison des scores R^2 obtenus pour toutes les méthodes.

Les modèles obtiennent des résultats différents suivant les jeux de données. Les méthodes **HSTT4**, **RF** et **LSTM** obtiennent les meilleurs résultats dans respectivement 14/39, 15/39 et 10/39 des cas. La méthode **HST-MLP** obtient des résultats qui sont toujours en dessous de ceux des autres méthodes.

Ces résultats peuvent s'expliquer en considérant que le modèle MLP est moins expressif que le modèle **LSTM**, et que **HST-MLP** utilise moins de variables d'entrées que **RF**, et est plus difficile à calibrer que **HSTT4**.

Ces résultats impliquent que ces modèles sont complémentaires : les meilleurs résultats sont obtenus soit par **HSTT4**, **RF** ou **LSTM**.

Cependant, ces 4 modèles ont des coûts de calculs très différents : **HSTT4** est de loin le plus rapide à calibrer et calculer (quelques secondes par jeux de données), alors que **HST-MLP**, **RF**, **LSTM** peuvent demander beaucoup de temps de calculs pour être entraînés :

- Pour l'apprentissage de **HST-MLP** : de 1 à 5 minutes suivant le jeu de données, en utilisant un GPU Nvidia GeForce 1080 Ti.
- Pour l'apprentissage de **RF** : de 1 à 10 minutes suivant le jeu de données en utilisant grid search et feature selection (RFE), sur un CPU AMD Ryzen 9 3900X (en utilisant un seul cœur par jeux de données). Seulement quelques secondes sans utiliser grid search et feature selection.
- Pour l'apprentissage des **LSTM** : de 5 minutes à 1-2 heures suivant le jeu de données en utilisant un GPU Nvidia GeForce 1080 Ti.

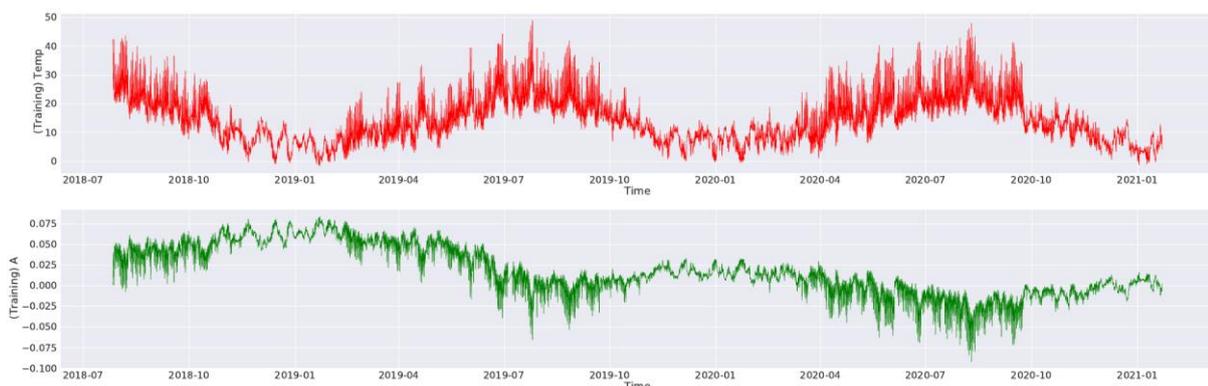
A noter : après avoir été entraînés, ces modèles ne demanderont que quelques secondes pour être appliqués à un jeu de données (comme **HSTT4**). La complexité des calculs réside principalement dans la phase d'apprentissage.

Les modèles **RF** et **HSTT4** sont donc ceux avec le meilleur compromis résultats vs temps de calculs.

Par la suite, nous nous intéresserons donc à l'application de ces modèles pour retirer les effets de température sur des signaux d'auscultation.

Pour ce faire, nous utiliserons les modèles HSTT4, HST-MLP, RF et LSTM comme modèles régressifs du signal mesuré à partir de la température, et calculerons un signal corrigé à partir des résidus de cette régression (méthodologie présentée dans les sections précédentes). Nous comparons les signaux corrigés suivant chaque modèle régressif.

Les Figures 5 et 6 présentent des exemples de ces résultats pour différents jeux de données, et en utilisant différentes méthodes de modélisation (RF, LSTM). On observe la présence d'anomalies dans les signaux corrigés des deux exemples : en septembre 2019 pour le premier exemple, et à fin 2013 pour le second exemple. Dans les deux cas, ces anomalies sont difficiles à identifier dans les signaux bruts, mais apparaissent clairement dans les signaux corrigés, les rendant beaucoup plus facile à traiter -- en particulier pour des méthodes de détection d'anomalies automatiques basés sur des dépassements de seuils.



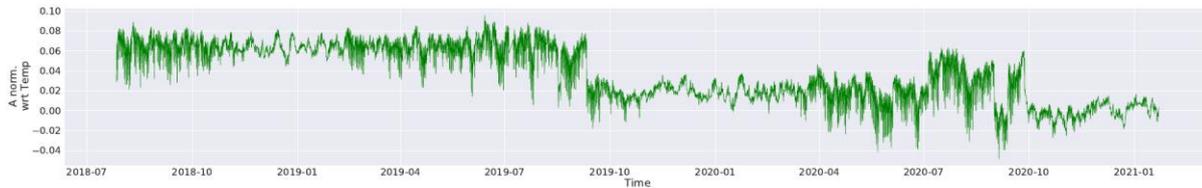


Figure 5. Jeu de données « Tilt 03 » issu d'un tiltmètre, modélisation à partir du modèle **RF**. Figure du haut (rouge) : signal de température (°C), figure du milieu (vert) : signal brut (tilt en degrés), figure du bas (vert) : signal corrigé sans effet de la température

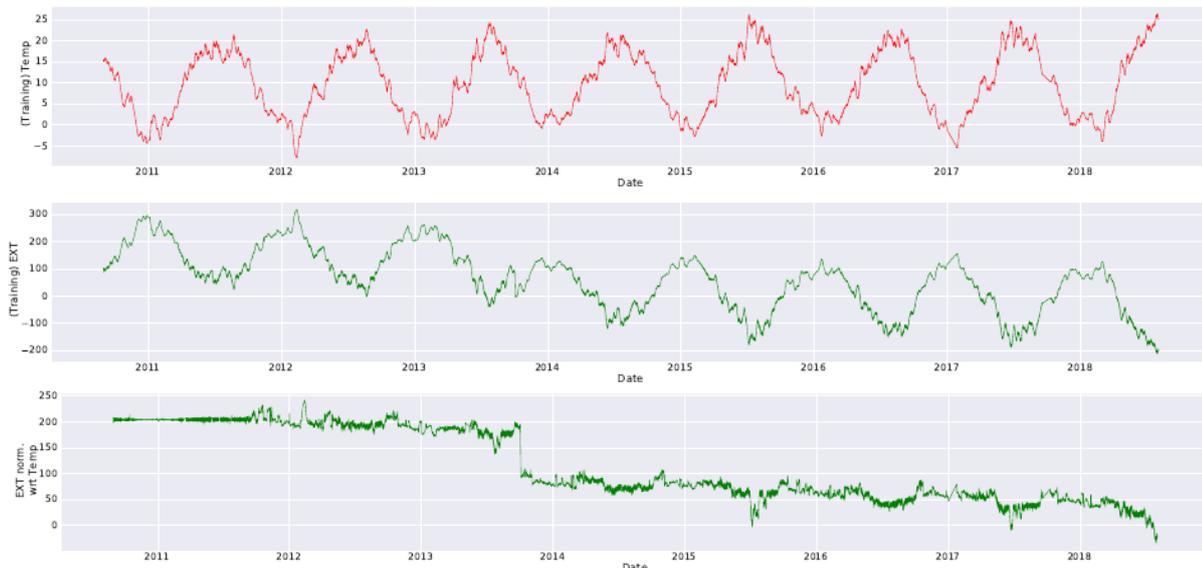


Figure 6. Jeu de données « Pont 1 » issu d'un extensomètre, modélisation à partir du modèle **LSTM**. Figure du haut : signal de température (°C), figure du milieu (vert) : signal brut (déplacement en mm), Figure du bas (vert) : signal corrigé sans effet de la température

2. Validation par simulation de dégradations

Il s'agit d'une piste de travail future.

Nous cherchons à valider notre approche sur une collection de jeux de données qui est représentative des mesures d'auscultation obtenues sur une panoplie d'ouvrages et de constructions différents. Cependant, il est difficile de regrouper suffisamment de jeux de données correspondant à toutes les configurations d'intérêt, ou qui représentent l'ensemble des dégradations possibles. Dans ce contexte, nous proposons de tester notre méthodologie de correction des effets environnementaux sur des jeux de données comportant des dégradations artificielles, que nous qualifierons de « dégradations simulées ». En particulier, cette approche permettrait de vérifier si notre méthodologie permettrait de retirer correctement le biais issu des effets de la température.

Notre approche pourra prendre en compte des combinaisons de deux types de dégradations simulées : soit des dégradations de « long terme », soit de « court terme ».

Les dégradations de long terme correspondent par exemple à des tassements. Ces dégradations D_t^l sont des processus asymétriques qui se produisent lentement sur de longues périodes de temps (au moins 30% de la durée du jeu de données). Pour les simuler, nous appliquerons la

formule suivante au signal brut s_t , entre deux pas de temps t_d et t_f pour obtenir le signal avec simulation de dégradations S_t^{lt} :

$$D_t^{lt} = \sum_{i=t_d}^t d_i^{1.7}$$

$$S_t^{lt} = s_t + D_t^{lt}$$

Où d_i est un vecteur contenant $t_f - t_d$ valeurs aléatoires tirées de la distribution normale $\mathcal{N}(0, \sigma^2)$, et avec σ la déviation standard de s_t entre les pas de temps de t_d à t_f .

Les dégradations de court terme D_t^{ct} correspondent à des « pics » (déplacements, etc.) du signal. Ils se produisent sur de courtes périodes de temps (au plus 5% de la durée du jeu de données). Pour les simuler et obtenir le signal dégradé S_t^{ct} , nous sélectionnons un pas de temps t_{pic} correspondant au “pic” de dégradation entre deux pas de temps t_d et t_f correspondant respectivement au début et à la fin de la dégradation du signal brut s_t .

On considère : $D_{t_{pic}}^{ct} = k$ avec k une variable aléatoire dans le domaine $[\min(s) + 0.75(\max(s) - \min(s)), \min(s) + 2.0(\max(s) - \min(s))]$.

Les valeurs de D_t^{ct} quand $t \neq t_{pic}$ sont obtenues par interpolation quadratique (cela permet donc d’avoir une courbe uniforme).

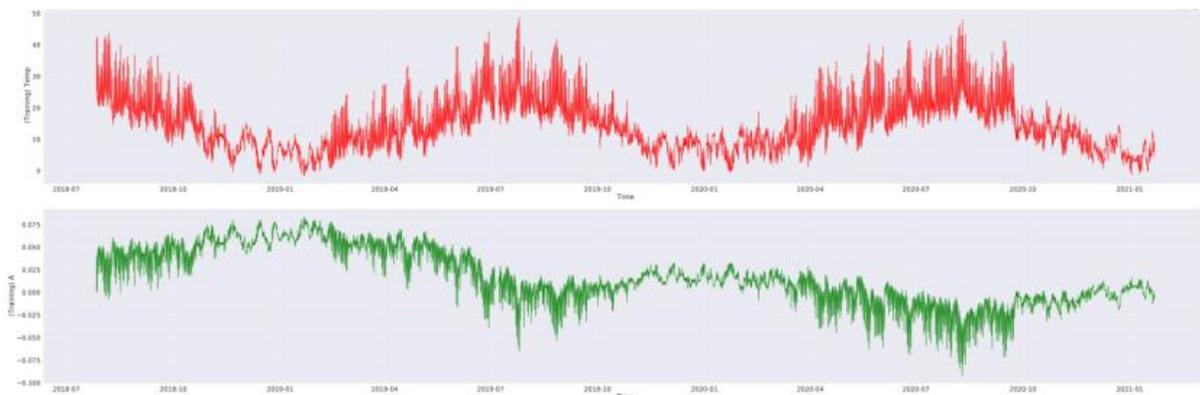
Finalement, on a :

$$S_t^{ct} = s_t + D_t^{ct}$$

Cette procédure peut être répétée plusieurs fois pour obtenir plusieurs « pics » entre les pas de temps t_d et t_f

Pour obtenir des collections de signaux contenant différentes dégradations simulées, nous appliquons (ou non) chacune de ces deux types de dégradations plusieurs fois pour obtenir chaque signal dégradé : entre 0 et 2 fois pour les dégradations de long terme, et entre 0 et 6 fois pour les dégradations de court terme.

La Figure 8 présente des exemples de signaux contenant des dégradations simulées, se basant sur le signal brut de la Figure 7 (milieu).



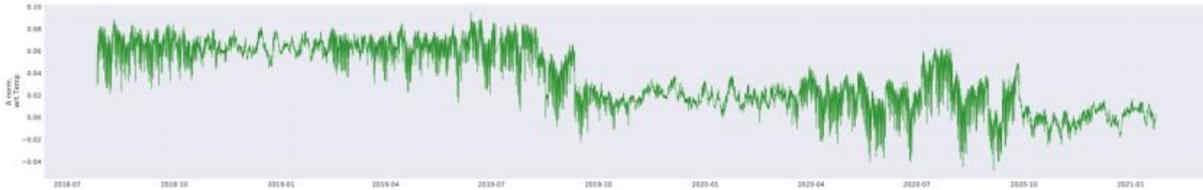


Figure 7. Signaux de référence : **haut** (rouge) correspond au signal de température, **milieu** (vert) représente le signal brut considéré, **bas** (vert) est le signal corrigé (sans effet de la température)

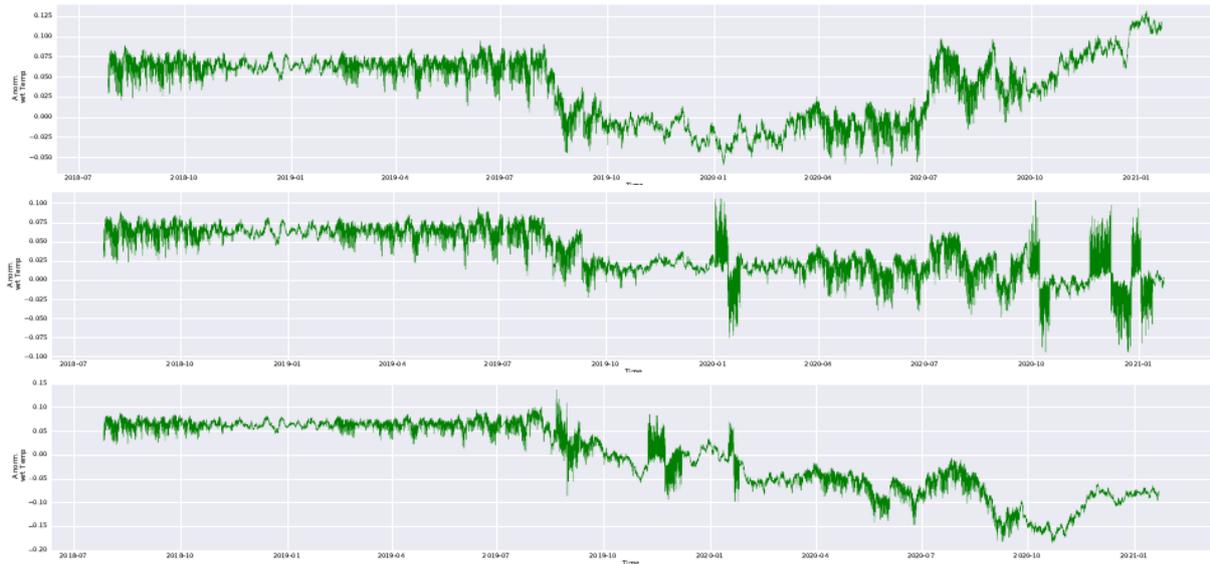


Figure 8. Exemples de signaux avec dégradations simulées se basant sur le signal brut de la Figure 7 (milieu) et contenant respectivement : **haut** seulement des dégradations de long terme, **milieu** seulement des dégradations de court terme, et **bas** plusieurs dégradations de longs et courts termes.

Nous prévoyons d'utiliser ces signaux contenant des dégradations simulées pour établir la performance de notre méthode de correction des effets de températures sur différents types de dégradations (capacité de généralisation). Ces travaux sont à réaliser en 2022.

3. Visualisation des degrés de dégradation

3.1 Méthode

La méthode, que l'on pourrait nommer « Securiscore » est une stratégie d'interprétation de dégradations (anomalies) pour le domaine de la surveillance de l'intégrité des structures et ouvrages. Cette méthode fonctionne en temps réel et permet de visualiser le degré de dégradation du système étudié : plus le feu est proche du vert, moins il y a d'anomalies détectées, et plus le feu est rouge, plus il y a d'anomalies. La méthode se base sur la déviation entre le signal prédit et le signal corrigé (où l'influence de la température a été retirée). Comme le signal prédit est obtenu à partir d'un modèle entraîné sur les conditions normales (sans anomalies), cela revient à comparer les déviations entre le signal en condition normale et le signal brut pendant la période surveillée.

Il convient de noter que comme cette méthodologie de correction se base sur les résidus (comme notre méthodologie de correction de l'effet de la température) provenant de modèles prédictifs, elle comporte une incertitude intrinsèque, en particulier parce que la modélisation du signal n'est pas « parfaite », et se base sur des bases d'apprentissages bruitées.

Des méthodes existent pour estimer le degré des incertitudes des modèles prédictifs.

La méthode « sécuriscore » compare les incertitudes sur le signal prédit avec le signal corrigé. On regarde si le signal corrigé se trouve bien dans les intervalles de confiance du signal prédit, où s'il les dépasse. Les intervalles de confiance font office de seuils au-delà desquels on a une détection d'anomalie (avec un niveau de sévérité qui augmente de seuil en seuil), et qui varient de pas de temps en pas de temps.

La méthode « sécuriscore » (SC) est généralement plus précise pour détecter des anomalies qu'une méthode à dépassement de paliers (DP) pour deux raisons :

1. SC fait une comparaison pour chaque pas de temps entre conditions normales et conditions observées, ce qui apporte plus de précision et flexibilité comparé à DP, où le palier est généralement constant.
2. SC prend en compte les incertitudes du modèle de correction des effets de la température, ce qui n'est pas le cas de DP.

Dans le cas de notre modèle SC, il est relativement simple de calculer des intervalles de confiance correspondant à l'incertitude de la prédiction -- par exemple en utilisant la méthode « Quantile Random Forests » [20].

La méthode des « Quantile Random Forests » se base sur des random forests classiques (donc un ensemble de Decision Trees). Au cours de l'apprentissage, ces arbres de décisions vont être tous entraînés pour cadrer aux données d'apprentissage de plusieurs façons possibles (chaque arbre est différent des autres). Les résultats de régression de chaque arbre vont être combinés (moyennes, votes, ou d'autres stratégies suivant l'algorithme) pour sortir le résultat de régression de la random forest. On peut donc aussi faire des statistiques sur la variance des résultats sur l'ensemble des arbres. L'idée générale de la méthode des « Quantile Random Forests » se base sur cette variance (et autres statistiques) pour calculer l'incertitude de la régression.

On applique un code couleur en fonction de l'écart entre le signal prédit et la déviation standard du signal prédit, pour montrer l'écart avec l'attendu, et donc le potentiel niveau de risque sur la mesure.

Nous utilisons la méthode des « Quantile Random Forests » pour calculer les incertitudes sous la forme de quantiles correspondant à la probabilité d'obtenir différents degrés de variation autour du signal corrigé.

Nous approximations la valeur de σ à partir de ces quantiles, en utilisant la formule suivante :

$$\sigma \approx 1.35 * IQR$$

avec IQR l'écart interquartile, c'est-à-dire la différence entre les percentiles 75% et 25%.

3.2 Résultats

Nous appliquons cette méthode de visualisation sur plusieurs jeux de données. Les résultats de plusieurs jeux de données d'exemples sont montrés en Figures 10 12 et 13.

Nous considérons l'exemple de la Figure 10 sur un extensomètre « Pont 1 (extensomètre) »:

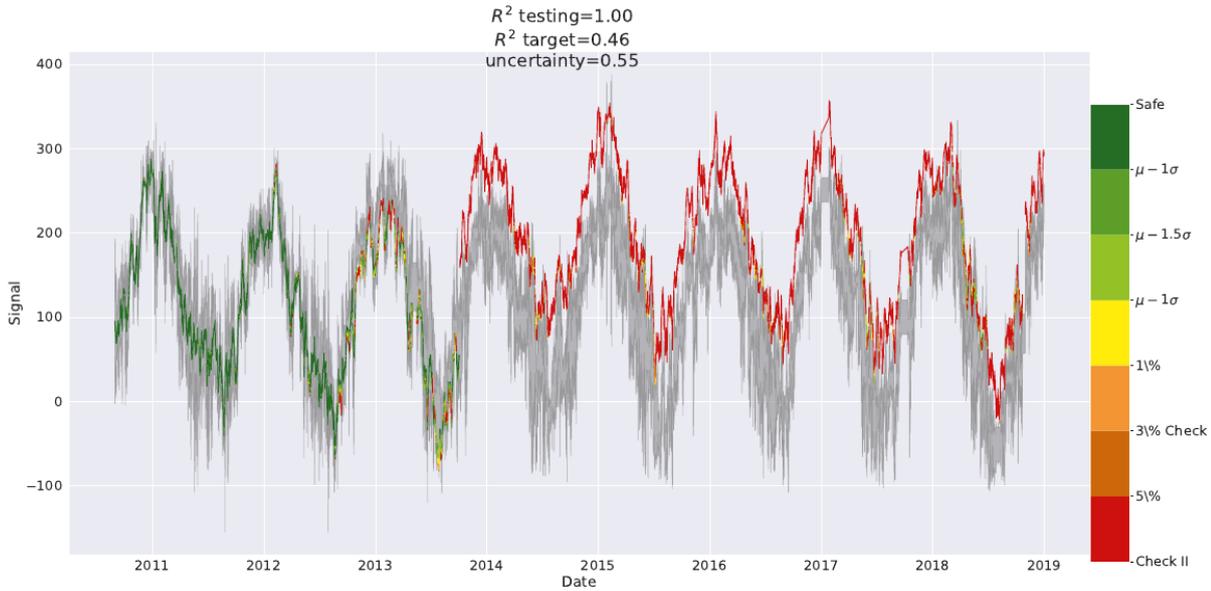


Figure 10. Exemple de la méthode sécuriscore sur le jeu de données Pont 1 (extensomètre)

On voit qu'il y a une anomalie de long-terme après la 3eme année. La courbe du signal corrigé avec l'effet de la température retiré est montrée dans la Figure 11 :

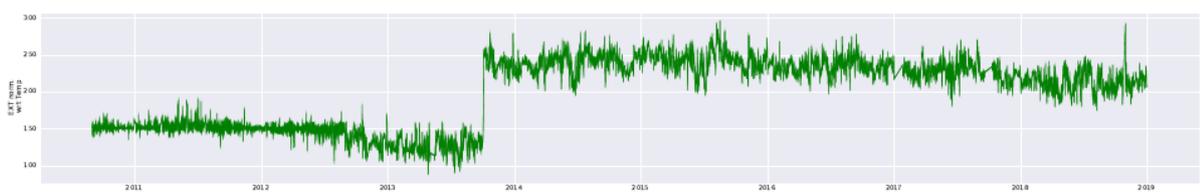


Figure 11. Signal corrigé du jeu de données Pont 1 (extensomètre)

On voit bien la présence d'un effet vers fin 2013, qui correspond, sur la Figure 8 au passage au "rouge" du système sécuriscore.

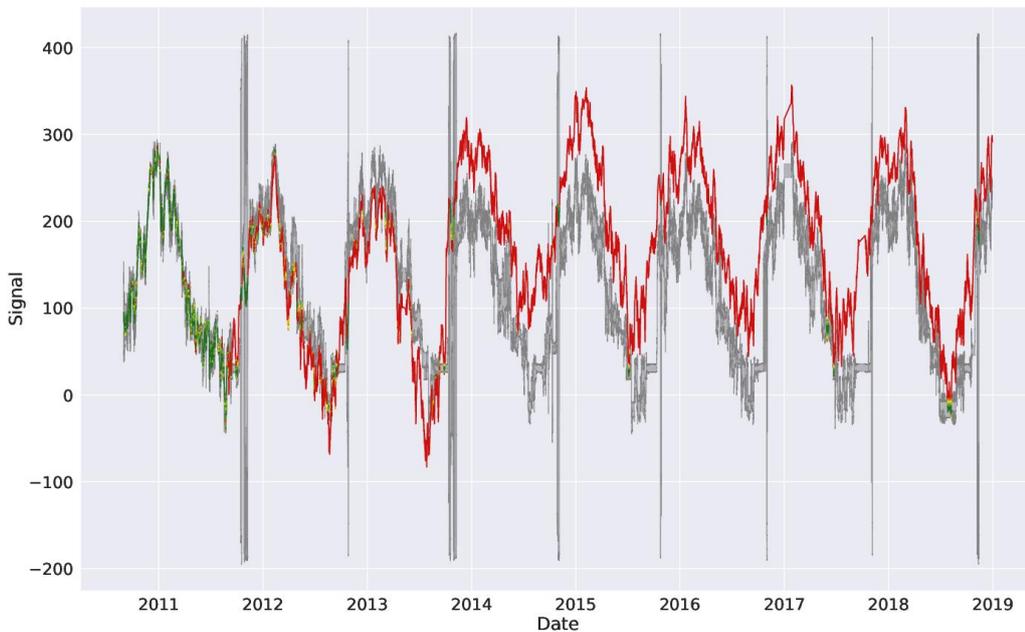


Figure 12. Exemple de la méthode sécuriscore sur le jeu de données Pont 2 (extensomètre)

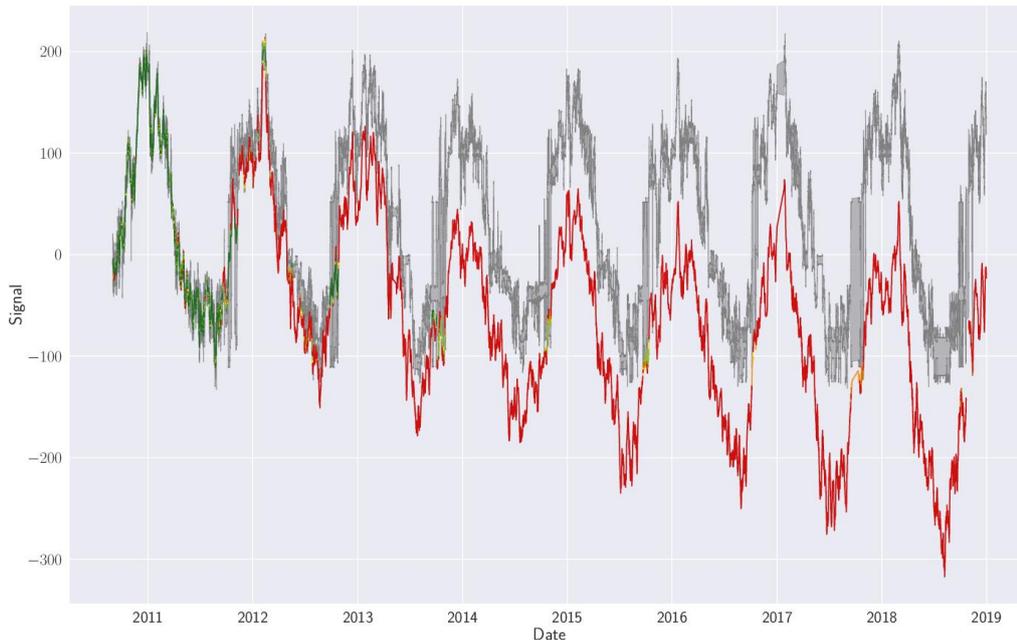


Figure 13. Exemple de la méthode securiscore sur le jeu de données Pont 3 (extensomètre)

Il serait possible d'adapter cette méthode aussi au modèle **HSTT** : pour cela, il faudrait calculer analytiquement l'incertitude (et donc changer les équations pour la prendre en compte). Cependant, il serait difficile d'obtenir des intervalles de confiance avec des méthodes basées sur des réseaux de neurones (comme **HST-MLP**, ou **LSTM**).

3.3 Discussions et conclusion

Notre étude se place dans le contexte de la surveillance de structures et d'ouvrages, et nous abordons le problème de la correction des biais issus de la température ambiante sur les mesures d'auscultation. Notre approche se base sur une méthodologie établie de suppression des biais [2, 14], et l'étend en proposant de nouvelles façons de modéliser les effets de températures.

Ces nouveaux modèles sont de plusieurs natures différentes. Tout d'abord, nous considérons le modèle **HST** (un modèle de référence pour l'auscultation des barrages) et sa variante **HSTT** (**HST** avec prise en compte des effets de température).

Par ailleurs, nous avons testé d'autres modèles issus de la communauté de l'apprentissage automatique (un sous-domaine de l'intelligence artificielle) : un modèle utilisant des forêts aléatoires (**RF**), et deux modèles utilisant des réseaux de neurones artificiels : le premier (**MLP-HST**) est tiré de la littérature [13, 14] et utilise un simple réseau « feed-forward », le second est introduit dans notre étude et utilise un réseau récurrent (**LSTM**).

Nos résultats montrent que les modèles **RF**, **LSTM** et **HSTT** donnent les meilleurs résultats de modélisation.

Nous illustrons ensuite, comment ces modèles prédictifs peuvent être utilisés pour corriger les biais issus de la température ambiante sur les bâtiments.

D'autre part, nous définissons une méthode pour valider ce système de correction des biais en quantifiant la performance des modèles sur des jeux de données réels ou artificiels. Dans le deuxième cas, nous générons des jeux de données contenant plusieurs dégradations simulées de deux types différents : long-terme et court-terme.

Finalement, nous proposons une méthode de visualisation des niveaux d'écart à la normale. Cette méthode permet de détecter facilement la présence d'anomalies dans les signaux corrigés en prenant aussi en compte les incertitudes sur le processus de modélisation du signal brut.

Notre étude pourrait encore être étendue en validant notre méthodologie de façon plus approfondie. En particulier, un de nos travaux en cours consiste à vérifier si la méthodologie de correction de température permet bien d'aboutir à des signaux corrigés dans lesquels les anomalies restent apparentes sans délai depuis leur apparition dans la structure. Cette étape de vérification s'effectue en utilisant des données artificielles, qui correspondent aux jeux de données introduits dans cette étude, mais modifiés pour introduire des dégradations à court ou long termes (pics, tassements irréversibles, etc.).

D'autre part, nous prévoyons d'étendre notre approche pour être capable d'entraîner des modèles généraux qui pourraient s'appliquer directement à des capteurs récemment installés sur une nouvelle structure, et ce, sans étape d'apprentissage. Cette opération pourra être rendue possible au travers d'une méthodologie de « transfert d'apprentissage » (« transfer learning » en anglais) dans laquelle un modèle sera entraîné sur plusieurs jeux de données pour devenir suffisamment général pour se généraliser à des nouvelles structures.

4. Références

1. De Bigault de Granrut M. Analyse et interprétation de la pression d'eau en fondation des barrages-voûtes à partir des mesures d'auscultation. Diss. Université Grenoble Alpes (ComUE), 2019.
2. Loose BR, Timothy S, and Strawn T. Instrument bias correction with machine learning algorithms: Application to field-portable mass spectrometry. *Frontiers in Earth Science* 8 (2020): 557.
3. Simon A, et al. Analysis and interpretation of dam measurements using artificial neural networks. *Proceedings of the 9th ICOLD European club symposium, Venice, Italy. 2013.*
4. González, WM, Rubén LB, and Joaquín AB. Temperature measurement assisted modal tracking of an instrumented building. *Engineering Structures* 233 (2021): 111907.
5. Wager S, Trevor H, and Bradley E. Confidence intervals for random forests: The jackknife and the infinitesimal jackknife. *The Journal of Machine Learning Research* 15.1 (2014): 1625-1651.
6. Simon A, Royer M, Mauris F, Fabre J. Analyse des mesures d'auscultation des barrages avec les réseaux de neurones. In *Proceedings of the 9th ICOLD European club symposium, Venice, Italy 2013.*
7. Kingma DP, Ba J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.* 2014 Dec 22.
8. Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, Lin Z, Desmaison A, Antiga L, Lerer A. Automatic differentiation in pytorch.
9. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research.* 2011 Nov 1;12:2825-30.
10. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural computation.* 1997 Nov 15;9(8):1735-80.
11. Ho TK. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition 1995 Aug 14 (Vol. 1, pp. 278-282).* IEEE.
12. Gahlouz M. Analyse et interprétation des données d'auscultation des barrages par le modèle HST (Doctoral dissertation, Université Mouloud Mammeri Tizi Ouzou).
13. Penot I, Fabre JP, Daumas B. Analyse et modélisation du comportement des ouvrages de génie civil par la prise en compte des températures de l'air : méthode HST thermique. In *Twenty-third congress on large dams 2009.*
14. Nelder JA, Mead R. A simplex method for function minimization. *The computer journal.* 1965 Jan 1;7(4):308-13.
15. Cho S, Yun CB, Lynch JP, Zimmerman AT, Spencer Jr BF, Nagayama T. Smart wireless sensor technology for structural health monitoring of civil structures. *Steel Structures.* 2008 Dec 1;8(4):267-75.

16. Breiman L. Arcing classifier (with discussion and a rejoinder by the author). The annals of statistics. 1998 Jun;26(3):801-49.
17. L'auscultation des barrages en terre : une nécessité (S. Bonelli, K. Radzicki, J. Szczesny, R. Tourment, H. Felix)
18. Meinshausen N, Ridgeway G. Quantile regression forests. Journal of Machine Learning Research. 2006 Jun 1;7(6).
19. Haykin S. Neural networks and learning machines, 3/E. Pearson Education India; 2010.