



Détection d'anomalies et identification de leurs précurseurs dans des larges collections de séries temporelles - Application à un matériel de centrales EDF

Paul Boniol
Université Paris Cité
LIPADE
Paris, France
boniol.paul@gmail.com

Mohammed Meftah
EDF R&D
PRISME
Chatou, France
mohammed.meftah@edf.fr

Emmanuel Remy
EDF R&D
PRISME
Chatou, France
emmanuel.remy@edf.fr

Themis Palpanas
Université Paris Cité, IUF
LIPADE
Paris, France
themis@mi.parisdescartes.fr

Abstract—Large collections of time series are becoming a reality in many domains. There is therefore a serious need in developing efficient techniques to analyze and process this type of data. Especially, anomaly detection represents a crucial issue in order to detect these anomalies as soon as possible to avoid any critical event. Therefore, in this work, we address the two following tasks: (i) unsupervised anomaly detection in time series; (ii) supervised identification of precursors. We study these two tasks through their application to an industrial use case related to an EDF power plant system.

Keywords—time series, anomaly detection, identification of precursors, machine learning, neural network.

Résumé—Les larges collections de séries temporelles deviennent une réalité dans un grand nombre de domaines. Il y a donc un intérêt et un besoin croissants de développer des techniques efficaces pour analyser et traiter ce type de données. La détection automatique d'anomalies dans les séries temporelles représente un enjeu crucial dans l'objectif de détecter ces anomalies le plus rapidement possible pour éviter la survenue de tout événement critique. Par conséquent, dans ce travail, nous traitons les deux objectifs suivants : (i) l'exploration non-supervisée de séries temporelles pour la détection rétrospective d'anomalies à partir d'une collection de séries temporelles ; (ii) l'explication de la classification d'anomalies connues dans les séries temporelles, afin d'identifier de possibles précurseurs. Nous étudions ces deux points à travers une application à un cas industriel concret relatif à un matériel de centrales électriques EDF.

Keywords—séries temporelles, détection d'anomalies, identification de précurseurs, apprentissage machine automatique, réseaux de neurones.

I. INTRODUCTION

Les larges collections de séries temporelles deviennent une réalité dans un grand nombre de domaines [1], comme la finance, les sciences de l'environnement, la médecine, les métiers du numérique, l'ingénierie ou l'industrie. Il y a donc

un intérêt et un besoin croissants de développer des techniques efficaces pour analyser et traiter ce type de données.

De manière informelle, une série temporelle est une séquence ordonnée dans le temps de points ou de valeurs, par exemple des mesures successives d'un paramètre physique issues d'un capteur de surveillance d'un système industriel. Une fois les séries collectées et disponibles, les utilisateurs ont souvent besoin de les étudier pour en extraire de la valeur et de la connaissance. Ces analyses peuvent être simples, comme sélectionner des fenêtres temporelles pour visualiser les séries et calculer des statistiques sur les valeurs, mais aussi complexes, comme rechercher des similarités entre des séries ou détecter des anomalies, souvent synonymes d'évolutions soudaines et inhabituelles possiblement non souhaitées, voire de dysfonctionnement, du système industriel étudié. Ce dernier type d'analyse représente un enjeu crucial pour des applications dans un large éventail de domaines partageant tous le même objectif [2], [3] : détecter les anomalies le plus rapidement possible pour éviter la survenue de tout événement critique et allonger la durée de fonctionnement des systèmes.

Par conséquent, dans ce travail de recherche, nous traitons les deux objectifs suivants : (i) l'exploration non-supervisée de séries temporelles pour la détection rétrospective d'anomalies à partir d'une collection de séries temporelles ; (ii) l'explication de la classification d'anomalies connues dans les séries temporelles, afin d'identifier de possibles précurseurs.

Nous introduisons d'abord le contexte industriel, ainsi que des définitions fondamentales liées aux séries temporelles et aux méthodes de détection d'anomalies. Nous présentons ensuite nos contributions scientifiques en suivant les deux axes mentionnés précédemment. Ainsi, nous décrivons premièrement une solution originale pour la tâche de détection non-supervisée de sous-séquences anormales dans

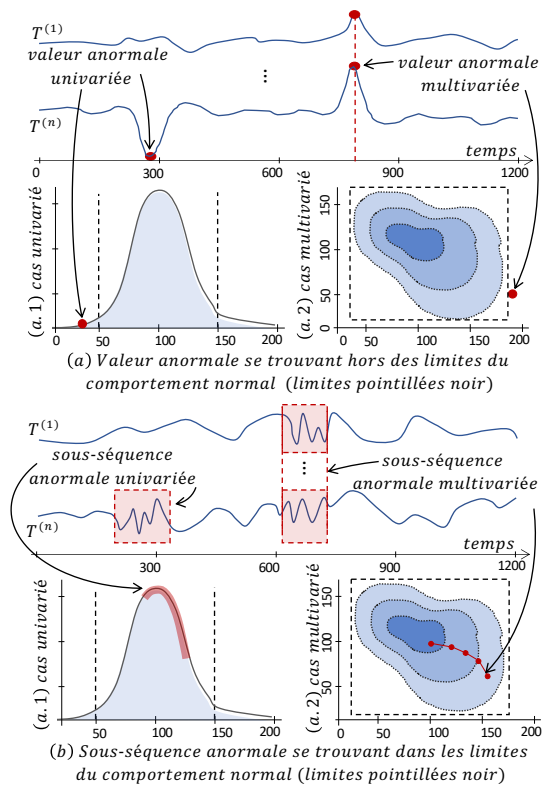


FIGURE 1. Exemples synthétiques où (a) un point aberrant (pour une série temporelle (a.1) univariée et (a.2) multivariée). (b) une sous-séquence anormale composée de valeurs individuellement normales (pour une série temporelle (b.1) univariée et (b.2) multivariée).

les séries temporelles [4]. Dans un deuxième temps, nous abordons le problème lié à l'identification supervisée de précurseurs d'anomalies. Nous subdivisons cette tâche en deux problèmes génériques : la classification supervisée de séries temporelles d'une part, l'explication des résultats de cette classification par l'identification de sous-séquences discriminantes d'autre part. Enfin, nous illustrons l'applicabilité et l'intérêt de nos développements au travers d'un cas d'usage portant sur l'identification de précurseurs de vibrations indésirables survenant sur des pompes de centrales électriques d'EDF.

II. CONTEXTE INDUSTRIEL ET DÉFINITIONS

A. Contexte industriel

Comme mentionné précédemment, le problème théorique de l'identification automatique d'anomalies répond à des besoins industriels réels. La possibilité d'analyser efficacement une grande quantité de données et d'être capable d'exprimer des requêtes complexes (comme par exemple la détection d'anomalies) peut être cruciale pour les acteurs industriels. EDF (l'un des principaux fournisseurs internationaux d'électricité) fait partie de ces acteurs. Un objectif important pour EDF est d'améliorer la sûreté et la disponibilité de ses centrales électriques en détectant, suffisamment tôt, les anomalies matérielles qui pourraient se produire, et ainsi éviter les défaillances. Il y a donc besoin de disposer d'algorithmes efficaces pour détecter les anomalies et comprendre leurs origines.

Par ailleurs, EDF collecte des données de capteurs dans chaque centrale nucléaire depuis au moins 20 ans. Sachant qu'un capteur peut renvoyer une valeur toutes les deux secondes, un historique complet de valeurs (c'est-à-dire une série temporelle) peut atteindre 315.000.000 points. Avec un total de 58 centrales nucléaires et avec plus de 2.000 capteurs par tranche, cela représente une base de séries de données de plus de 33 trillions de points. Sachant qu'une valeur nécessite 16 bits, une estimation grossière indique qu'une telle base de données correspond à 500 Téraoctets. Ainsi, pour tirer profit de ces données collectées dans le passé, il faut disposer de méthodes efficaces pour traiter ces larges séries temporelles. Dans la section suivante, nous décrivons et énumérons les méthodes existantes pour la détection de sous-séquences anormales en prenant soin d'étudier leur applicabilité au contexte des bases de données EDF. De plus, malgré le travail considérable effectué par les experts, les annotations d'anomalies dans les séries de données sont, soit inexistantes, soit difficiles à traiter manuellement ou automatiquement par un code informatique. Ainsi, il est nécessaire de s'intéresser à la fois aux méthodes supervisées (i.e. nécessitant des exemples d'anomalies et de comportements normaux), semi-supervisées (i.e. ne nécessitant que des exemples de comportements normaux), mais aussi non-supervisées (i.e. ne nécessitant aucun exemple).

B. Séries temporelles

Nous définissons une série temporelle *univariée* comme une séquence ordonnée de valeurs réelles sur une seule dimension. Par exemple, une série de données *univariée* peut correspondre à l'historique des valeurs d'un capteur. Dans ce cas, une sous-séquence (i.e. un extrait de points successifs de cette série) peut être représentée comme un vecteur de valeurs. Ensuite, nous définissons une série temporelle *multivariée* comme étant soit un ensemble de séquences ordonnées de valeurs réelles (chaque séquence ordonnée ayant la même longueur), soit une séquence ordonnée de vecteurs composés de valeurs réelles. Par exemple, une série temporelle *multivariée* peut être un ensemble de mesures issues de plusieurs capteurs installés sur un même système ou sur différents équipements. Dans ce cas précis, une sous-séquence est une matrice dans laquelle chaque ligne correspond à une sous-séquence d'une seule dimension.

Formellement, nous définissons une série temporelle univariée (i.e. sur une seule dimension) $T \in \mathbb{R}^n$ comme une séquence ordonnée de valeurs $T_i \in \mathbb{R} [T_0, T_2, \dots, T_{n-1}]$, pour laquelle $n = |T|$ est la taille (ou longueur) de la série T , et T_i est le $i^{\text{ème}}$ point de T .

Une série temporelle multivariée, ou multi-dimensionnelle, $\mathbf{T} \in \mathbb{R}^{(D,n)}$ est un ensemble de D séries temporelles de taille n . Nous notons cette dernière $\mathbf{T} = [T^{(0)}, \dots, T^{(D-1)}]$ et pour $j \in [0, D-1]$, nous notons la série temporelle univariée $T^{(j)} = [T_0^{(j)}, T_1^{(j)}, \dots, T_{n-1}^{(j)}]$. Une sous-séquence $T_{i,\ell}^{(j)} \in \mathbb{R}^\ell$ de la dimension $T^{(j)}$ de la série temporelle \mathbf{T} est un sous-ensemble continu de valeurs de taille ℓ (avec souvent $\ell \ll n$) commençant à la position i . Nous notons formellement une sous-séquence de la manière suivante : $T_{i,\ell}^{(j)} = [T_i^{(j)}, T_{i+1}^{(j)}, \dots, T_{i+\ell-1}^{(j)}]$. La sous-séquence multivariée

est définie comme suit : $T_{i,\ell} = [T_{i,\ell}^{(0)}, \dots, T_{i,\ell}^{(D-1)}]$. Soit une série temporelle T univariée, l'ensemble de toutes les sous-séquences possibles de T de taille ℓ est noté $\mathbb{T}_\ell = \{T_{0,\ell}, T_{1,\ell}, \dots, T_{|T|-\ell,\ell}\}$.

C. Détection d'anomalies

1) *Une première définition* : Tout d'abord, il convient de noter qu'il n'existe pas de définition unique, universelle et précise caractérisant les valeurs aberrantes ou les anomalies. En général, une anomalie est une observation qui semble s'écarter de façon notable des autres membres de l'échantillon dans lequel elle se produit. Cet écart peut indiquer que l'observation spécifique a été générée par un mécanisme différent de celui du reste des données (définition de Hawkins [5]). Ce mécanisme peut être une procédure erronée de mesure et de collecte de donnée ou une variabilité inhérente au domaine des données examinées. Néanmoins, de telles observations sont intéressantes dans les deux cas, et l'analyste gagnerait à les connaître. En pratique, la définition générale ci-avant peut prendre différentes formes, en fonction du problème spécifique et des caractéristiques des données manipulées. Par exemple, en statistiques, les anomalies peuvent être les points qui s'écartent de la moyenne de la distribution connue des données d'un certain nombre de fois l'écart-type. Ce cas implique d'effectuer des tests pour trouver la distribution (et ses paramètres) qui correspond le mieux à l'ensemble des données. Néanmoins, dans plusieurs problèmes et cas d'applications, nous ne connaissons pas précisément cette distribution. Par conséquent, définir et identifier les anomalies en utilisant leur distance par rapport à une valeur moyenne définie par un expert est parfois peu pratique.

2) *Point versus sous-séquence* : À ce stade, nous sommes intéressés par la recherche d'anomalies dans les séries temporelles. Cet objectif peut être atteint en examinant, soit des valeurs prises séparément, soit une séquence de points successifs (i.e. une sous-séquence). Dans le contexte spécifique des points, nous sommes intéressés par la recherche de ceux éloignés de la distribution des valeurs représentant le comportement "normal". Dans le contexte spécifique des séquences, nous sommes intéressés par l'identification de sous-séquences anormales qui, contrairement aux valeurs aberrantes, ne sont pas une valeur anormale unique, mais une évolution anormale de ces valeurs. Dans certains cas, cette distinction entre point et sous-séquence devient cruciale pour la raison suivante : même si chaque point pris individuellement semble normal, la forme générée par la séquence de ces mêmes valeurs peut être anormale et peut conduire à des dysfonctionnement qui n'auraient été détectés que trop tard. La Figure 1 illustre cette distinction entre point et sous-séquence anormaux. Dans ce travail, nous nous intéresserons au cas spécifique de la détection de sous-séquences anormales dans les séries temporelles.

III. DÉFINITION DES PROBLÈMES

Nous considérons désormais les deux cas de figure suivants : (i) on ne dispose pas d'informations précises (voire aucune) sur les anomalies à détecter (cadre non-supervisé). (ii) on

dispose d'exemples précis d'anomalies déjà survenues qu'on doit apprendre à détecter et comprendre (cadre supervisé).

A. Détection non-supervisée de sous-séquences anormales

Dans le premier cas (i), on peut décider d'adopter une démarche entièrement non-supervisée. Ces méthodes présentent l'avantage de ne pas avoir besoin d'une grande collection d'anomalies connues et de détecter automatiquement des comportements anormaux inconnus. Elles peuvent être utilisées, soit pour surveiller en temps réel l'état de santé d'un système, soit pour exploiter rétrospectivement des séries temporelles déjà collectées d'un système (afin de constituer une collection de comportements anormaux qui peuvent ensuite être utilisés dans un cadre supervisé). La Figure 2 illustre cette tâche à l'aide d'un cas d'application relatif à une centrale électrique, dans lequel la Figure 2(a)(1) présente les séries temporelles issues des capteurs de sous-systèmes de la centrale (Figure 2(b)). L'objectif est de trouver a posteriori des sous-séquences anormales inconnues dans les séries temporelles (illustrées par le rectangle rouge dans la Figure 2(a)(1)). Nous proposons maintenant une définition formelle du problème.

Définition 1 (Problème de détection non-supervisée): Soit un système M et un ensemble de séries temporelles \mathbf{T}_M qui représente l'état de santé de M . On doit trouver la fonction f qui prend comme entrées \mathbf{T}_M et renvoie $s \in \{\mathcal{N}, \mathcal{A}\}$ (où \mathcal{N} correspond à l'état sain, tandis que \mathcal{A} indique une anomalie). Ainsi, f peut être définie comme suit : $f : \mathbf{T}_M \rightarrow \{\mathcal{N}, \mathcal{A}\}$.

B. Identification supervisée de précurseurs

Dans le premier cas (i), les anomalies sont considérées comme inconnues. Dans le second cas (ii), nous considérons qu'on connaît précisément l'événement à détecter et qu'on dispose d'une collection de séries temporelles correspondant à ces anomalies. Dans ce cas, on dispose donc d'une base de données d'anomalies. Par conséquent, on peut décider d'utiliser des méthodes supervisées. À ce stade, la question qui se pose est la suivante : est-il possible de détecter des sous-séquences qui se sont produites avant l'anomalie connue et qui pourraient l'expliquer (et potentiellement la prédire). De telles sous-séquences peuvent être appelées "précurseurs" d'anomalies.

La détection de telles sous-séquences est d'une grande utilité pour les experts afin de prévenir l'apparition de futures anomalies ou de comprendre pourquoi une anomalie s'est produite. Ainsi, la tâche consiste à détecter, de manière supervisée, les anomalies connues et à retrouver leurs précurseurs potentiels. La Figure 2 illustre cette tâche sur un cas d'application concernant une centrale électrique. Plus précisément, la Figure 2(a)(2) illustre les séries temporelles de capteurs provenant des sous-systèmes de la centrale (Figure 2(b)). L'objectif est de trouver des précurseurs inconnus (T_1 et T_2) de sous-séquences anormales dans les séries temporelles (illustrées par le rectangle rouge dans la Figure 2(a)(2)). Nous proposons maintenant une définition formelle de ce problème.

Définition 2 (Problème d'identification de précurseurs): Soit un système M , un ensemble de séries temporelles $\mathbf{T}_M^{\mathcal{N}}$ qui

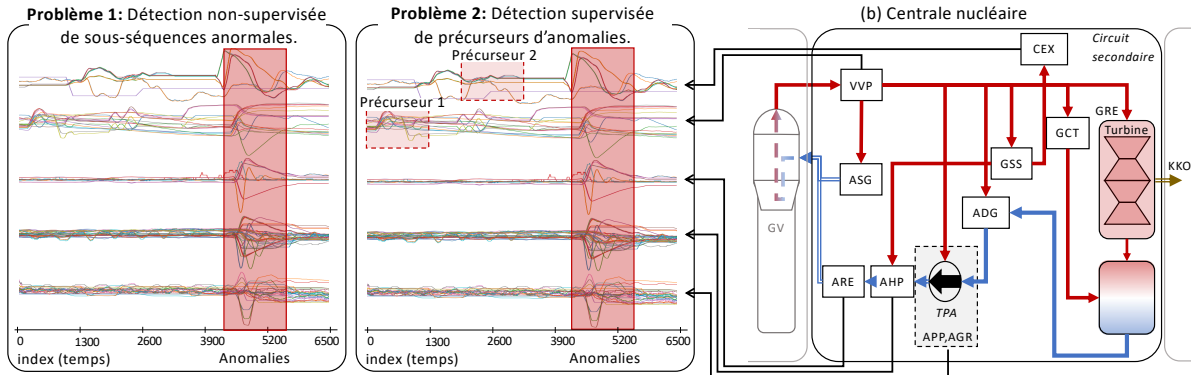


FIGURE 2. Problèmes de détection de sous-séquences anormales et identification de leurs précurseurs dans les cas où : (a) les séries temporelles sont issues de capteurs provenant (b) de sous-systèmes d'une centrale nucléaire.

représente l'état normal de M (état correspondant au label \mathcal{N}) et un ensemble de séries temporelles \mathbf{T}_M^A qui représente l'état de M avant un état anormal (état correspondant au label \mathcal{A}), il faut d'abord trouver la fonction f qui prend en entrée \mathbf{T}_M^A et \mathbf{T}_M^N et retourne $s \in \{\mathcal{N}, \mathcal{A}\}$. Il faut ensuite trouver la fonction g qui prend en entrée \mathbf{T}_M^A et f et retourne $S \subset \mathbf{T}_M^A$ (S étant l'ensemble des sous-séquences dans \mathbf{T}_M^A précurseurs des anomalies à venir). Formellement, f et g peuvent être écrites : $f : \mathbf{T}_M^N, \mathbf{T}_M^A \rightarrow \{\mathcal{N}, \mathcal{A}\}$ et $g : \mathbf{T}_M^A, f \rightarrow S$.

IV. SOLUTIONS PROPOSÉES

Nous synthétisons d'abord brièvement les approches de l'état de l'art pour chacun des deux objectifs. Nous présentons ensuite nos contributions pour résoudre ces problèmes.

A. Détection non-supervisée de sous-séquences anormales

1) *Limitations des travaux existants* : Certaines des techniques existantes recherchent explicitement un ensemble de types d'anomalies prédéterminés, comme RUSboost [6], ou identifient comme anomalies les sous-séquences présentant les plus grandes distances par rapport à leurs plus proches voisins (appelés *discords*), comme Matrix Profile [3] et GrammarViz [7]. Ces approches posent des limites à la tâche d'identification des sous-séquences anormales pour plusieurs raisons.

Premièrement, le comportement anormal n'est pas toujours connu. Par conséquent, les techniques qui utilisent des connaissances spécifiques à un domaine pour identifier les anomalies (par exemple en cardiologie [6] ou en ingénierie [8]) impliquent plusieurs paramètres finement estimés et ne sont pas généralisables à de nouvelles applications.

Deuxièmement, en ce qui concerne les techniques généralisables de détection de sous-séquences anormales, les algorithmes de pointe (par exemple Matrix Profile [3] et GrammarViz [7]) sont efficaces surtout dans le cas où une seule anomalie, ou plusieurs anomalies différentes les unes des autres, se trouve dans les séries temporelles. Cela s'explique par le fait que ces algorithmes utilisent comme fondement la distance des sous-séquences par rapport à leur voisin le plus proche dans la série temporelle : la sous-séquence qui a le voisin le plus éloigné est marquée comme une anomalie. Par

conséquent, une anomalie qui se répète (au moins deux fois) ne sera pas identifiée.

Afin de remédier à cette limitation, des approches utilisant le principe de m^{th} discord ont été proposées [9]. Ces algorithmes prennent en compte la multiplicité, m , des sous-séquences anormales qui sont similaires les unes aux autres, et marquent comme anomalies toutes les sous-séquences du même groupe en calculant les $m^{èmes}$ (au lieu du 1^{er}) plus proches voisins pour chaque sous-séquence. Néanmoins, ces techniques supposent de connaître la multiplicité m , ce qui n'est pas facile en pratique. En effet, si m est inconnue, nous devons exécuter plusieurs fois l'algorithme pour plusieurs valeurs différentes de m , ce qui peut impliquer un temps de calcul très important.

Finalement, un autre inconvénient des méthodes non-supervisées existantes pour la détection de sous-séquences anormales est la non-stationnarité des séries temporelles : les caractéristiques des données (les statistiques de base et les tendances) peuvent changer au fil du temps. Ces situations sont difficiles à gérer et rendent confuses les méthodes *discord* et m^{th} -*discord*. Cette difficulté s'explique par le fait qu'une sous-séquence anormale peut trouver un voisin très proche parmi les sous-séquences d'une autre partie de la série temporelle étant statistiquement différente.

2) *Series2Graph* : Nous présentons désormais Series2Graph [4], une méthode de détection de sous-séquences anormales basée sur des graphes orientés pour des séries temporelles univariées. Notre approche ne nécessite pas d'instances étiquetées (comme les techniques supervisées), ni de données propres ne contenant pas d'anomalie. Intuitivement, le graphe construit par Series2Graph encode toutes les sous-séquences d'une série temporelle (unique ou faisant partie d'une collection de séries) et les comportements potentiellement récurrents de ces sous-séquences. Ceci permet de différencier le comportement normal, c'est-à-dire les motifs qui se produisent fréquemment, des anomalies, c'est-à-dire les sous-séquences qui reviennent rarement dans la série temporelle. Ce mode de représentation a l'avantage d'être insensible à la cardinalité et à la similarité des anomalies. De plus, il permet d'utiliser le même modèle (i.e. graphe) pour la détection de sous-séquences de longueurs différentes. Nous décrivons maintenant les principales étapes de calcul

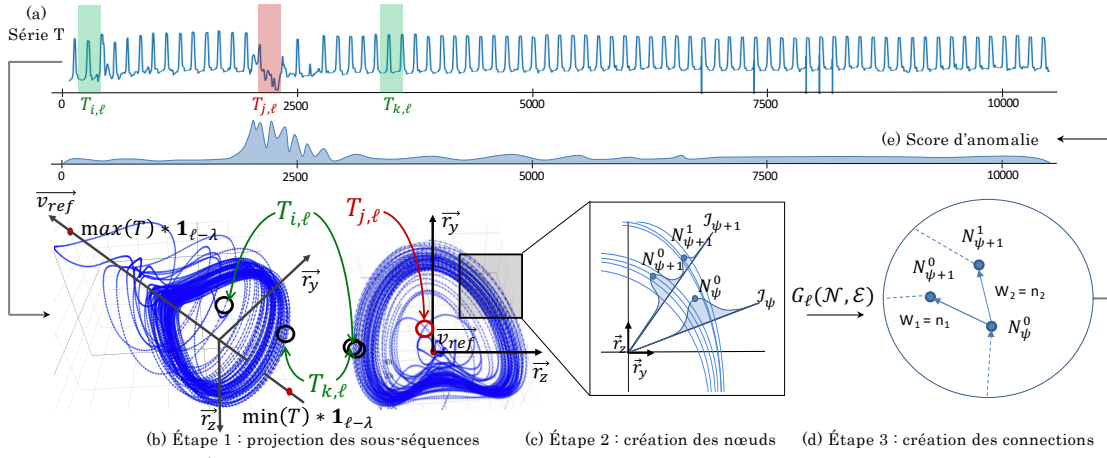


FIGURE 3. Étapes de construction du graphe par la méthode Series2Graph représentant la série temporelle.

de Series2Graph, les détails de la méthode étant présentés dans [4].

- **[Projection des sous-séquences]** : Comme l'illustre la Figure 3(b), nous projetons toutes les sous-séquences (d'une longueur donnée ℓ) de la série T dans un espace tridimensionnel correspondant aux trois composantes les plus importantes de l'analyse en composantes principales de l'ensemble des sous-séquences. Cet espace est ensuite transformé en un espace bidimensionnel, constitué de deux composantes \vec{r}_y, \vec{r}_z correspondant à deux vecteurs orthogonaux de v_{ref} , un axe composé de chaque séquence plate ($a * \mathbf{1}_{\ell-\lambda}$ avec $a \in \mathbb{R}$). Dans ce dernier espace en deux dimensions, la similarité de forme est préservée. Par exemple, la Figure 3 représente trois sous-séquences $T_{i,\ell}, T_{j,\ell}$ et $T_{k,\ell}$. $T_{i,\ell}$ et $T_{k,\ell}$ ont la même forme et donc le même emplacement dans l'espace bidimensionnel, alors que $T_{j,\ell}$ (qui a une forme différente) ne l'est pas.
- **[Création des nœuds]** : Nous créons ensuite un nœud pour chacune des parties les plus denses de l'espace bidimensionnel (voir Figure 3(c)). Pour ce faire, l'espace est d'abord discrétisé en un ensemble de rayons \mathcal{J}_{ψ} d'angles ψ . Nous estimons ensuite la densité des sous-séquences le long de chacun de ces rayons en utilisant des noyaux gaussiens. Les valeurs maximales des densités estimées sont attribuées aux nœuds et forment notre ensemble de nœuds \mathcal{N} . Ces nœuds résument tous les principaux motifs de longueur ℓ qui se sont produits dans T .
- **[Création des connexions]** : Nous récupérons ensuite toutes les transitions entre les paires de sous-séquences représentées par deux nœuds différents : chaque transition correspond à une paire de sous-séquences, où l'une apparaît immédiatement après l'autre dans la série T . Comme le montre la Figure 3(d), nous représentons chaque transition par une connexion entre les nœuds correspondant, et nous formons ainsi l'ensemble des connexions \mathcal{E} . Le poids d'une connexion est défini par le nombre de fois où la paire de sous-séquences correspondante a été observée dans T . Enfin, nous construisons notre graphe $G_{\ell}(\mathcal{N}, \mathcal{E})$.

- **[Calcul du score des sous-séquences]** : Nous calculons le score de normalité (ou d'anomalie) d'une sous-séquence de longueur $\ell_q \geq \ell$ (provenant ou non de T), en nous basant sur les connexions/nœuds et leurs poids/degrés (le degré d'un nœud correspondant au nombre de connexions entrantes et sortantes). Formellement, pour une sous-séquence T_{i,ℓ_q} de T , représentée par un chemin dans $G_{\ell}(\mathcal{N}, \mathcal{E})$, noté $P_{th} = \langle N^{(i)}, N^{(i+1)}, \dots, N^{(i+\ell_q)} \rangle$, le score de normalité est défini par : $Norm(P_{th}) = \sum_{j=i}^{i+\ell_q-1} \frac{w(N^{(j)}, N^{(j+1)}) (deg(N^{(j)}) - 1)}{\ell_q}$, où $w(e)$ et $deg(n)$ sont respectivement le poids de la connexion e et le degré du nœud n . Enfin, nous prenons l'inverse de ce score de normalité pour calculer le score d'anomalie (illustré par la Figure 3(e)). Ce dernier est utilisé pour identifier les sous-séquences anormales (comme par exemple $T_{k,\ell}$ dans la Figure 3(a)).

Afin de démontrer l'efficacité de Series2Graph, nous avons effectué une évaluation approfondie à l'aide de plusieurs jeux de données de la littérature provenant de différents domaines [4]. Nous avons évalué aussi sa capacité à identifier des anomalies de tailles différentes. Cette expérimentation démontre la supériorité de Series2Graph en termes de précision de détection d'anomalies, ainsi qu'une certaine flexibilité à la taille des sous-séquences recherchées.

B. Identification supervisée de précurseurs

Nous décrivons maintenant nos contributions liées à la tâche d'identification supervisée des précurseurs d'anomalies (problème décrit dans la Définition 2). Nous traitons celle-ci comme une tâche de classification dans laquelle l'anomalie est une classe spécifique et les précurseurs des caractéristiques discriminantes cette classe.

De façon générale, la classification de séries temporelles est un problème crucial et difficile en science des données [10]. Pour résoudre ce problème, divers algorithmes ont été proposés au cours des dernières années [11], appliqués à un grand nombre de cas d'usage. Les méthodes standard sont basées sur les distances aux plus proches voisins. La classification k-NN (utilisant les distances euclidiennes ou DTW (Dynamic Time

Warping)) en est un exemple [12]. Néanmoins, des travaux récents ont montré que les méthodes d'ensemble utilisant des classificateurs plus avancés atteignent de meilleures performances [13]. Suite aux récentes percées dans le domaine de la vision assistée par ordinateur, de nouvelles études proposent avec succès des méthodes d'apprentissage profond pour la classification de séries temporelles [14]–[17], tels que les réseaux de neurones convolutifs (CNN), les réseaux neurones résiduels (ResNet) [18] et InceptionTime [19].

De manière synthétique, un CNN prend en entrée une série temporelle multivariée \mathbf{T} , dont chaque dimension correspond à une série temporelle univariée issue d'un capteur. Il est composé de plusieurs couches convolutives. Une couche convolutive, notée $conv(n_f, \ell)$, est définie par un nombre de kernels n_f . Un kernel est défini par une taille ℓ donnée, et est constitué de poids et biais $\mathbf{W}, \mathbf{B} \in \mathbb{R}^{(D, \ell)}$, D étant le nombre de dimensions de la série temporelle multivariée en entrée \mathbf{T} . Plus généralement, le résultat de $conv(n_f, \ell)$ est une série temporelle multivariée avec n_f dimensions et de longueur n (n étant la longueur de la série temporelle multivariée). Un kernel va donc glisser le long de la série temporelle d'entrée et agit comme un filtre. Les kernels vont sélectionner au fur et à mesure des couches convolutives les patterns d'intérêts pour classifier la série temporelle d'entrée. L'apprentissage d'un CNN (i.e., calage des poids et biais \mathbf{W}, \mathbf{B}) utilise une fonction de perte mesurant le taux de classification correcte (dans ces travaux nous utilisons la cross-entropy comme fonction de perte). Cette fonction de perte est ensuite calculée pour différents batches (i.e. ensemble de séries temporelles multivariées en entrée) pour optimiser les poids et biais du CNN (l'optimiseur ADAM [20] a été utilisé ici). Le processus est répété de manière itérative (une itération est appelée epoch), jusqu'à la convergence de la valeur de la fonction de perte.

Même si on dispose d'un modèle de classification correctement entraîné, la recherche des explications du résultat de la classification (c'est-à-dire la recherche des caractéristiques discriminantes qui ont permis au modèle de décider de la classe à attribuer à chaque série temporelle multivariée en entrée) est un problème difficile. Il en reste néanmoins important, comme par exemple en médecine pour la formation des chirurgiens assistés par robot [21]. Ces caractéristiques discriminantes peuvent correspondre à des motifs d'intérêt qui se produisent dans un sous-ensemble de dimensions, à différents (ou au même) instants. Pour certains modèles de réseaux de neurones convolutifs, la Class Activation Map (CAM) [22] peut être utilisée pour expliquer le résultat de la classification. Par exemple, CAM a été proposée pour mettre en évidence les parties d'une image qui contribuent le plus à la prédiction d'une classe donnée. Cette méthode a ensuite été adaptée aux séries temporelles [14], [18].

Définissons maintenant CAM [14]. Pour une série temporelle \mathbf{T} , soit $A(\mathbf{T})$ le résultat de la dernière couche convolutive $conv(n_f, \ell)$. $A_m(\mathbf{T})$ est donc la série temporelle univariée pour la dimension $m \in [1, n_f]$ correspondant au noyau m . Soit $w_m^{C_j}$ le poids entre le noyau m et le

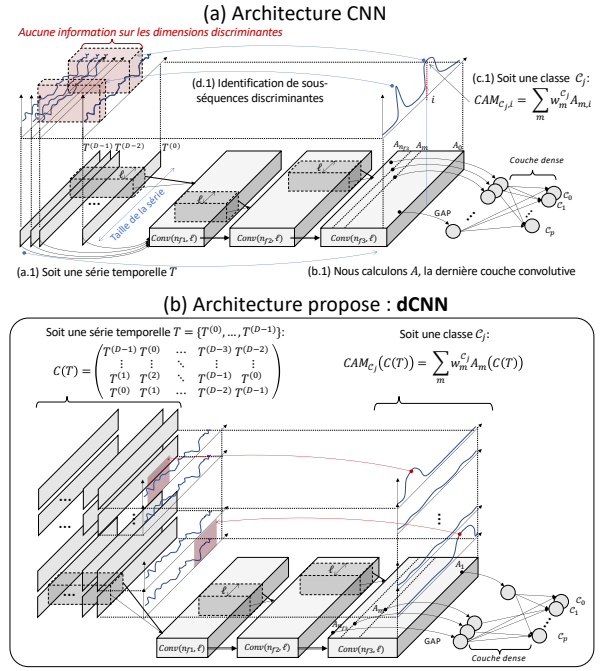


FIGURE 4. Architectures (a) CNN et (b) dCNN

neurone de sortie de la classe $C_j \in \mathcal{C}$. $CAM_{C_j}(\mathbf{T}) = [CAM_{C_j, 0}(\mathbf{T}), \dots, CAM_{C_j, n-1}(\mathbf{T})]$, qui souligne les caractéristiques discriminantes de la classe C_j , est définie comme suit :

$$\forall i \in [0, n-1], CAM_{C_j, i}(\mathbf{T}) = \sum_m w_m^{C_j} A_{m, i}(\mathbf{T}).$$

De manière informelle, CAM correspond à une moyenne pondérée des résultats des différents kernels. Plus le poids $w_m^{C_j}$ correspondant au kernel m est élevé, plus celui-ci a contribué à l'attribution à la classe C_j . Nous pouvons donc considérer que les poids forts $A_{m, i}(\mathbf{T})$ correspondent à des instants temporels i importants.

Par conséquent, $CAM_{C_j}(\mathbf{T})$ est une série temporelle univariée où chaque élément à l'indice i indique son importance (indépendamment des dimensions) pour la classification en classe C_j . La Figure 4(a) illustre le processus de calcul de CAM et de recherche des sous-séquences discriminantes dans la série initiale.

1) *Limitations des travaux existants* : Néanmoins, CAM appliquée aux séries temporelles souffre d'une limitation importante. En effet, CAM est une série temporelle univariée (de la même longueur n que les instances d'entrée) avec des valeurs élevées alignées sur les sous-séquences de la série temporelle multivariée en entrée qui contribuent le plus à l'identification d'une classe donnée. Cependant, dans le cas spécifique de séries temporelles multivariées, CAM ne donne aucune information sur le niveau de contribution individuel de chaque dimension. Résoudre cette limitation importante est un défi qui permettrait, par exemple, de déterminer à la fois la période temporelle et les dimensions (i.e. les capteurs) qui pourraient être liées à une anomalie potentielle.

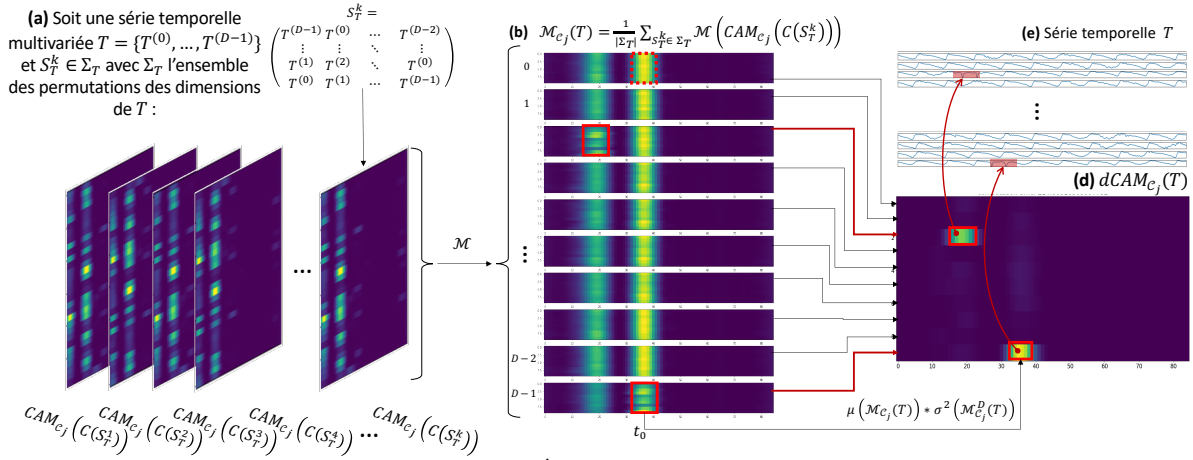


FIGURE 5. Étapes de calcul de dCAM

2) *Architecture dCNN*: Afin de contourner cette limite, nous proposons l'architecture dCNN [23], où l'entrée est transformée en un cube, dans lequel chaque ligne contient une combinaison donnée de toutes les dimensions de la série temporelle multivariée d'origine. Un noyau (de taille $(D, \ell, 1)$) glisse ensuite sur toutes les dimensions D fois, permettant à l'architecture d'apprendre des caractéristiques sur plusieurs dimensions simultanément. De plus, la CAM résultant est alors une série temporelle multivariée : une de ses lignes correspond à une combinaison donnée des dimensions. Cependant, nous cherchons toujours à récupérer des informations pour chaque dimension séparément. Pour ce faire, nous nous assurons que chaque ligne du cube d'entrée contient une permutation différente des dimensions. Comme les poids des noyaux correspondent à des positions fixes (pour des dimensions spécifiques), une permutation des dimensions donnera une CAM différente. Formellement, pour une série \mathbf{T} , on note $C(\mathbf{T}) \in \mathbb{R}^{(D,D,n)}$ la structure de données d'entrée du dCNN. $C(\mathbf{T})$ est définie dans la Figure 4(c). Notez qu'une dimension donnée $T^{(i)}$ n'est jamais à la même position dans les lignes de $C(\mathbf{T})$, cette dernière propriété étant cruciale pour le calcul de dCAM (la variante de CAM pour l'architecture dCNN).

La Figure 4 représente l'architecture dCNN. L'entrée $C(\mathbf{T})$ est transmise à un CNN classique en deux dimensions. Le reste de l'architecture est indépendant de la structure des données d'entrée, ce qui signifie que toute autre architecture peut être utilisée (comme ResNet). De même, la procédure d'apprentissage peut être choisie librement par l'utilisateur.

3) *Calcul de dCAM* : À ce stade, nous avons notre réseau pour classer les instances parmi les classes $\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_p$. Présentons maintenant comment calculer dCAM [23] qui identifiera les caractéristiques discriminantes parmi les D dimensions d'une série temporelle multivariée en entrée de longueur n . Nous supposons que le réseau dCNN entraîné est suffisamment performant en termes de classification pour fournir une dCAM qui ait du sens [23].

À première vue, nous pouvons calculer CAM pour la classe \mathcal{C}_j de la manière suivante : $CAM_{\mathcal{C}_j}(C(\mathbf{T})) = \sum_m w_m^{\mathcal{C}_j} A_m(C(\mathbf{T}))$. Cependant, une valeur élevée à la ligne i en position t sur $CAM_{\mathcal{C}_j}(C(\mathbf{T}))$ ne signifie pas que la

sous-séquence en position t de la dimension i est importante pour la classe \mathcal{C}_j . Cela signifie plutôt que la combinaison des différentes dimensions à la ligne i de $C(\mathbf{T})$ est importante.

Étant donné que différentes combinaisons de dimensions (c'est-à-dire une ligne de $C(\mathbf{T})$) produisent différentes sorties, les positions des dimensions dans les lignes de $C(\mathbf{T})$ ont donc un impact sur CAM. Nous définissons maintenant les étapes nécessaires pour la construction de dCAM.

- **Calcul des permutations** : Nous créons d'abord N_p permutations aléatoires des D dimensions de \mathbf{T} . Nous notons la $k^{\text{ème}}$ permutation S_T^k . Ensuite, pour une classe donnée \mathcal{C}_j , nous calculons la transformation $\mathcal{M}(CAM_{\mathcal{C}_j}(C(S_T^k))) \in \mathbb{R}^{(D,D,n)}$ (avec $CAM_{\mathcal{C}_j}(C(S_T^k)) \in \mathbb{R}^{D,n}$ et $CAM_{\mathcal{C}_j}(C(S_T^k))_i$ sa ligne i), définie pour une position p et une dimension d données, par :

$$\mathcal{M}(CAM_{\mathcal{C}_j}(C(S_T^k)))_{d,p} = CAM_{\mathcal{C}_j}(C(S_T^k))_{idx(T^{(d)}, p)}$$

Dans cette dernière équation, $idx(T^{(d)}, p)$ renvoie les indices de ligne dans $C(\mathbf{T})$ qui contiennent la dimension $T^{(d)}$ à la position p . Globalement, la transformation \mathcal{M} enrichit CAM en ajoutant l'information sur la position de la dimension. Ainsi, le calcul de $\mathcal{M}(CAM_{\mathcal{C}_j}(C(\mathbf{T})))$ pour différentes permutations de dimensions de \mathbf{T} fournira des informations distinctes concernant l'importance d'une position donnée dans une dimension donnée.

- **Assemblage des permutations** : Nous fusionnons maintenant les N_p permutations. Nous calculons la matrice $\bar{\mathcal{M}}_{\mathcal{C}_j}(\mathbf{T})$ de toutes les transformations \mathcal{M} de ces permutations :

$$\bar{\mathcal{M}}_{\mathcal{C}_j}(\mathbf{T}) = \frac{1}{|\Sigma_T|} \sum_{S_T^k \in \Sigma_T} \mathcal{M}(CAM_{\mathcal{C}_j}(C(S_T^k)))$$

La Figure 5 illustre le processus de calcul de $\bar{\mathcal{M}}_{\mathcal{C}_j}(\mathbf{T})$ à partir de Σ_T , l'ensemble des permutations de \mathbf{T} . $\bar{\mathcal{M}}_{\mathcal{C}_j}(\mathbf{T})$ peut être considéré comme un résumé de l'importance de chaque dimension à chaque position dans $C(\mathbf{T})$, pour toutes les permutations calculées.

- **Extraction de dCAM** : Nous pouvons désormais utiliser $\bar{\mathcal{M}}_{\mathcal{C}_j}$ calculé à l'étape précédente pour extraire des informations explicatives sur les sous-séquences considérées

comme importantes par le réseau dCNN. Tout d’abord, remarquons que chaque ligne de $C(\mathbf{T})$ correspond au format d’entrée de l’architecture CNN standard. Par conséquent, nous pouvons supposer que $\mu(\bar{\mathcal{M}}_{C_j}(\mathbf{T})) = \sum_{d \in [0, D-1]} \sum_{p \in [0, D-1]} \bar{\mathcal{M}}_{C_j}^{d,p}(\mathbf{T}) / (2 * D)$ est équivalent à $CAM_{C_j}(\mathbf{T})$. Par ailleurs, en plus de l’information temporelle toutes dimensions confondues, nous pouvons extraire une information temporelle par dimension. En effet, nous savons que pour une position donnée p et une dimension donnée d , $\bar{\mathcal{M}}_{C_j}^{d,p}(\mathbf{T})$ représente l’activation moyenne pour un ensemble donné de permutations. Si l’activation $\bar{\mathcal{M}}_{C_j}^{d,p}(\mathbf{T})$ pour une dimension donnée est constante (quelle que soit sa valeur, ou la position p), alors la position de la dimension d n’est pas importante, et aucune sous-séquence de cette dimension d n’est discriminante. D’autre part, une valeur élevée ou faible à une position spécifique p signifie que la sous-séquence à cette position est discriminante. S’il est naturel d’interpréter une valeur élevée, l’interprétation d’une valeur faible est contre-intuitive. Habituellement, une sous-séquence à la position p avec une faible valeur devrait être considérée comme non discriminante. Néanmoins, si l’activation est faible pour p et élevée pour les autres positions, alors la sous-séquence en position p est la conséquence de la faible valeur et est donc discriminante. Nous pouvons donc extraire les sous-séquences significatives par dimension en calculant la variance de toutes les positions d’une dimension donnée, permettant ensuite de filtrer les fenêtres temporelles non pertinentes en utilisant la moyenne $\mu(\bar{\mathcal{M}}_{C_j}(\mathbf{T}))$. Formellement, pour une dimension donnée d , un indice temporel t et une classe donnée C_j :

$$dCAM_{C_j}(\mathbf{T})_{a,t} = \sigma^2(\bar{\mathcal{M}}_{C_j}^d(\mathbf{T})_t) * \mu(\bar{\mathcal{M}}_{C_j}(\mathbf{T})_t)$$

Comme illustré par la Figure 5(d) et la Figure 6(2)(c), dCAM prend la forme d’une carte d’activation, dans laquelle les sections en jaune correspondent aux sections de la série temporelle multivariée (i.e. sous-séquences dans des dimension spécifique) en entrée considérées importantes pour la classification par le réseau dCNN (les sections non importantes sont en bleu).

Il est important de noter que nous réalisons ces étapes en utilisant le réseau dCNN sans l’entraîner pour chaque permutation. Nous n’entraînons le réseau qu’une seule fois avant de lancer le calcul de dCAM.

Afin de démontrer la performance de dCNN et dCAM, nous avons effectué une évaluation approfondie à l’aide de plusieurs jeux de données provenant de différents domaines [23]. Cette expérimentation numérique a démontré l’efficacité de dCNN pour classifier les séries temporelles multivariées, ainsi que celle de dCAM pour identifier les facteurs discriminants.

V. APPLICATION À UN MATÉRIEL DE CENTRALES EDF

Illustrons maintenant l’applicabilité industrielle et l’intérêt de nos méthodes. Pour cela, nous allons étudier la découverte de possibles précurseurs de vibrations indésirables se produisant sur des turbopompes alimentaires des centrales nucléaires EDF. Ces pompes (deux pompes différentes par centrale,

notées TPA1 et TPA2) ont pour but d’augmenter la pression de l’eau (de 1 à 80 bar) avant de la faire passer dans le générateur de vapeur (à une pression de 80 bar). Des vibrations peuvent se produire et elles sont considérées comme problématiques lorsque la position de la pompe varie de quelques microns. Un capteur booléen est activé lorsque cela arrive. Ainsi, les experts sont intéressés pour savoir s’il existe des précurseurs possibles de ces vibrations indésirables dans d’autres capteurs situés à proximité de la pompe.

A. Base de données

À ce stade, nous devons créer une base de données de séries temporelles anormales (c’est-à-dire contenant des événements de vibration) et de séries temporelles sans anomalie. En suivant la suggestion de l’expert matériel EDF, nous avons sélectionné $D = 120$ capteurs installés sur 12 sous-systèmes de la centrale nucléaire. La Figure 2(b) résume les sous-systèmes analysés. Nous avons collecté toutes les vibrations indésirables qui se sont produites sur chaque centrale nucléaire française du palier 1300MW. Au total, nous avons 444 événements de vibrations. Nous créons ensuite nos séries temporelles multivariées en sélectionnant les mesures de chaque capteur entre 75 minutes avant les vibrations et 5 minutes après. Nous réglons le taux d’acquisition à 1 point toutes les 6 secondes (soit une longueur de séries temporelles de $n = 800$ points). Notons \mathbf{T}_M^A l’ensemble de ces séries temporelles contenant une vibration. Pour compléter la base de données, nous sélectionnons ensuite 444 intervalles de 80 minutes pour lesquels aucune vibration n’a été enregistrée au moins un jour avant et après. Notons \mathbf{T}_M^N l’ensemble de ces séries temporelles sans aucune vibration. Nous avons également fait en sorte que les périodes sans vibration soient dans les mêmes conditions de fonctionnement que les périodes de vibration. Ces conditions de fonctionnement se produisent lorsque le réacteur monte ou descend en puissance entre 15% et 67% de la puissance maximale. Cette zone, où est couplée la deuxième pompe d’eau d’alimentation, est propice aux vibrations. Cette dernière étape est importante car, sans elle, les précurseurs mis en évidence risqueraient d’être la différence déjà connue des conditions opérationnelles et sollicitations. À cette fin, nous avons utilisé le capteur lié au régime de puissance et fait en sorte que sa distribution soit la même dans les deux classes \mathbf{T}_M^N et \mathbf{T}_M^A . Nous avons également pris la même distribution des années pour les deux classes, afin de minimiser l’influence d’une éventuelle dégradation due au vieillissement. Ce que nous voulons mettre en évidence, ce sont les sollicitations inattendues qui conduiraient à des vibrations. Nous disposons donc au total de 888 séries temporelles multivariées (dont la moitié correspond à des vibrations indésirables) de 120 dimensions et de longueur 800 chacune. Au total, les séries de données contiennent 85 248 000 points. Formellement, nous définissons l’ensemble de données comme $\mathbf{T}_M = \mathbf{T}_M^N \cup \mathbf{T}_M^A$, avec $\mathbf{T}_M^N, \mathbf{T}_M^A \in \mathbb{R}^{(444, 120, 800)}$.

La tâche consiste à détecter correctement les événements de vibration et à découvrir des sous-séquences, dans un ou plusieurs capteurs, qui se sont produites avant la vibration et qui

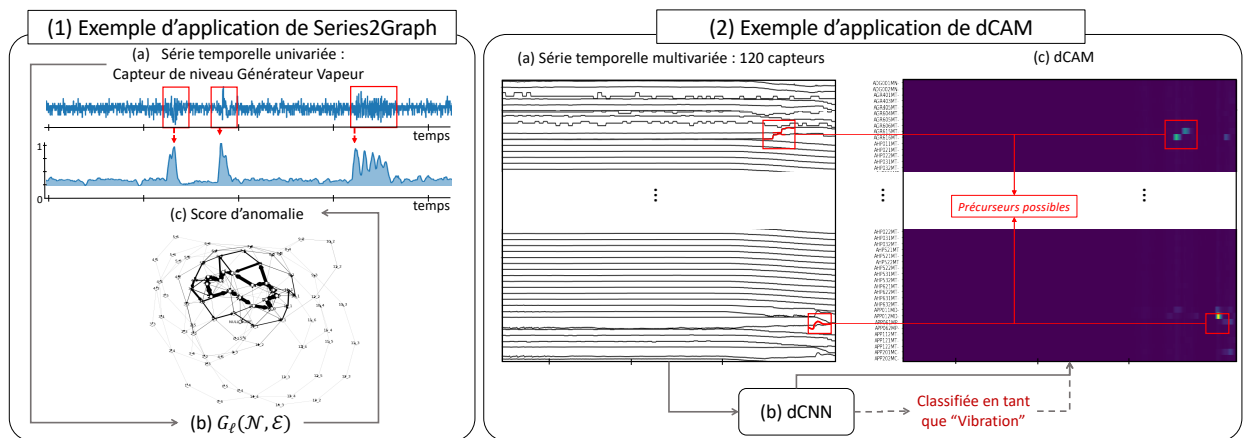


FIGURE 6. Exemple d'application de Series2Graph et dCAM sur des séries temporelles issues de capteurs de centrales EDF.

pourraient potentiellement l'expliquer. Pour cela, nous abordons cette tâche comme définie dans le Problème 1 avec Series2Graph, et NormA [24], une autre approche non-supervisée (détectant les anomalies avec leur distance à une partition de clusters de sous-séquences) que nous avons développé et qui a démontré aussi une grande précision pour détecter les sous-séquences anormales. Nous abordons ensuite cette tâche comme définie dans le Problème 2 avec dCNN/dCAM. Nous évaluons et comparons ainsi les performances des méthodes supervisées et non supervisées sur un même cas d'application.

B. Détection non-supervisée de vibrations

1) *Calcul du score d'anomalie* : Nous commençons par analyser comment les approches non-supervisées, telles que NormA et Series2Graph, peuvent détecter les séries temporelles présentant une vibration. Nous illustrons l'application de Series2Graph dans la Figure 6(1). Comme ces méthodes sont non-supervisées, nous construisons les modèles sur des séries de données avec et sans anomalies. Nous définissons les paramètres tels que définis dans [4] pour Series2Graph et [24] pour NormA, et nous utilisons une longueur de sous-séquence de 20 points. Comme les méthodes ne peuvent être utilisées que pour des séries temporelles univariées, nous construisons un modèle par capteur. Nous calculons ensuite une série temporelle de score d'anomalie pour chaque série temporelle. Nous notons $Score_{NormA}(T^{(d)})$ (avec $|Score_{NormA}(T^{(d)})| = |T^{(d)}|$) le score d'anomalie de la série de données $T^{(d)}$ en utilisant le modèle NormA entraîné sur la dimension d (avec $0 \leq d < D$) et $Score_{S2G}(T^{(d)})$ le score d'anomalie de $T^{(d)}$ en utilisant le modèle Series2Graph entraîné sur la dimension d .

Nous calculons ensuite la valeur maximale de $Score_{S2G}(T^{(d)})$ et de $Score_{NormA}(T^{(d)})$ et l'utilisons comme score unique pour $T^{(d)}$. Nous nous attendons donc à ce que, pour deux séries temporelles $\mathbf{T}, \mathbf{T}' \in \mathbf{T}_M^A, \mathbf{T}_M^N$, il existe une dimension $0 \leq d < D$ telle que $max(Score_{NormA}(T^{(d)})) > max(Score_{NormA}(T'^{(d)}))$ et $max(Score_{S2G}(T^{(d)})) > max(Score_{S2G}(T'^{(d)}))$. Nous évaluons maintenant cette hypothèse.

2) *Résultats* : Nous évaluons maintenant formellement la précision de NormA et Series2Graph. Pour cela, nous utilisons

les trois mesures de performance (couramment utilisées dans la littérature) suivantes :

- **AUC-ROC** : Nous calculons le taux de vrais positifs et le taux de faux positifs pour 100 seuils différents entre la valeur minimale et la valeur maximale de tous les scores. À partir de ces derniers, nous obtenons une courbe appelée courbe ROC. Nous utilisons ensuite l'aire sous cette courbe comme mesure de la précision. Plus l'AUC-ROC est élevée, meilleur est le score d'anomalie.
- **AUC-PR** : Nous utilisons la précision (taux de vrais positifs divisé par la somme des taux de vrais positifs et faux négatifs) et le rappel (taux de vrais positifs divisé par la somme des taux de vrais positifs et faux négatifs) comme mesures de performance. Nous les calculons pour un ensemble de 100 seuils différents entre les valeurs minimale et maximale de tous les scores. Nous obtenons ainsi une courbe appelée courbe PR. Nous utilisons ensuite l'aire sous cette courbe comme mesure de la précision. Plus l'AUC-PR est élevée, meilleur est le score d'anomalie.
- **Taux de classification correcte** : Nous définissons le taux de classification correcte comme le nombre de séries temporelles correctement classées parmi toutes les instances. Nous utilisons donc un seuil et assignons à la classe vibration toutes les instances dont le score est supérieur à ce seuil. Nous calculons la précision de classification pour 100 seuils différents entre les scores minimum et maximum de tous les scores. Nous mesurons le taux de réussite maximal pouvant être obtenu avec un seuil.

La Figure 7 résume les mesures de performance obtenues en utilisant Series2Graph (en vert) et NormA (en rouge) pour chaque capteur. Pour la majorité des 120 capteurs, Series2Graph et NormA ne permettent pas de détecter les vibrations (i.e. faible AUC-ROC, AUC-PR et taux de classification correcte). Cependant, certains capteurs fournissent une précision relativement plus élevée. Par exemple, la plus grande précision est obtenue avec le capteur AGR412MT en utilisant Series2Graph pour les deux critères AUC-ROC (0.63) et AUC-PR (0.66) et pour le taux de réussite maximal réalisable (0.63). Pour NormA, la plus grande performance est obtenue avec différents capteurs en fonction de la mesure de

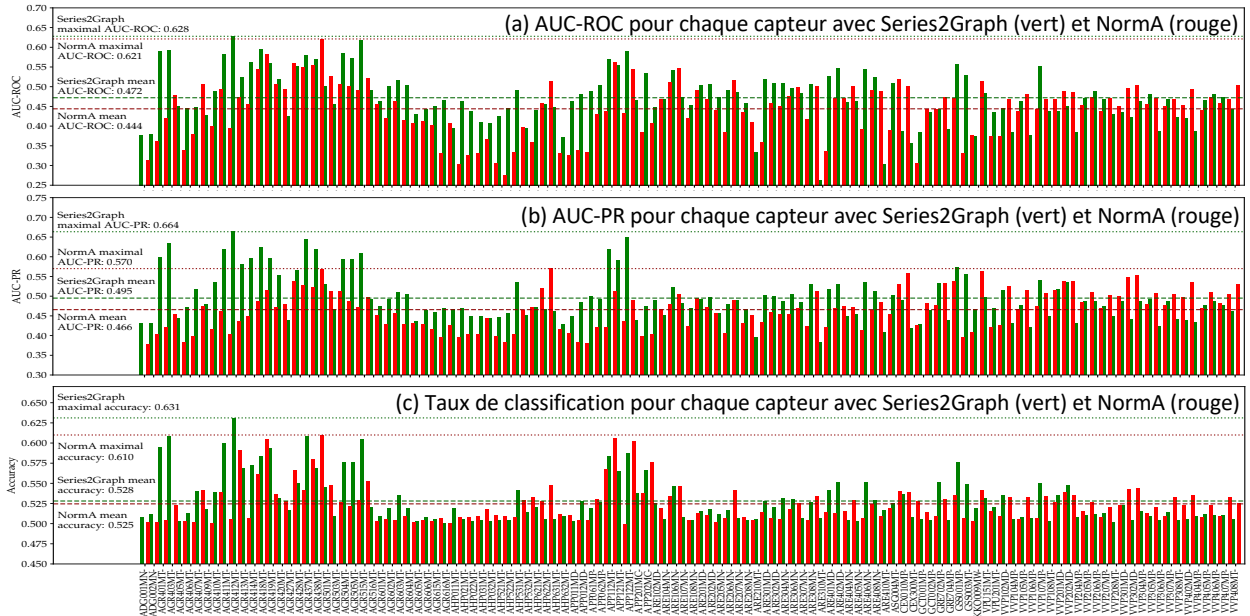


FIGURE 7. Taux de réussite de NormA et Series2Graph par capteur avec les critères (a) AUC-ROC, (b) AUC-PR, (c) taux de classification correcte.

précision utilisée. Par exemple, l’AUC-ROC (0.62) et le taux de réussite (0.61) maximaux de NormA sont obtenus avec le capteur AGR438MT, alors que l’AUC-PR (0.57) maximal est obtenu avec le capteur AHP622MT.

Dans l’ensemble, la Figure 7(c) montre que deux groupes de capteurs fournissent une précision légèrement supérieure à celle du reste des capteurs. Le premier groupe est le système de lubrification et de fluide de contrôle de la turbopompe (AGR), et le second groupe est le système de pompe à eau d’alimentation entraîné par la turbine (APP). Ils correspondent à des systèmes très proches et dépendants de la pompe vibrante et sont confirmés par l’expert comme étant physiquement (c’est-à-dire contenant des informations fonctionnelles des pompes telles que les températures d’étanchéité et le débit de la pompe de sortie) et structurellement cohérents avec la tâche de détection des vibrations.

C. Identification supervisée de précurseurs de vibrations

Nous évaluons maintenant la détection des vibrations à l’aide d’un modèle supervisé. Nous utilisons dCNN comme modèle supervisé dans cette section.

1) *Apprentissage* : Nous entraînons le modèle dCNN (en utilisant 5 couches convolutives avec les filtres (64, 128, 256, 256), une taille de noyau de 3 et un padding de 2) sur 70% des séries temporelles multivariées de chaque classe (\mathbf{T}_M^N et \mathbf{T}_M^A) et nous utilisons les 30% restants comme ensemble de validation. Nous fixons la taille des batches à 8 instances. Nous arrêtons la phase d’apprentissage de dCNN lorsque la fonction de perte sur l’ensemble de validation ne diminue pas au cours des 100 dernières itérations. Au total, nous limitons la phase d’apprentissage à 1000 itérations. Globalement, dCNN atteint une précision de classification de 0.91 sur l’ensemble d’entraînement et de 0.89 sur l’ensemble de validation (alors que NormA et Series2Graph fournissent une précision de 0.63 et 0.61, respectivement). Nous

confirmons donc qu’un modèle supervisé est significativement plus précis qu’un modèle non supervisé pour le cas de la détection de vibrations. La précision de dCNN étant élevée, nous pouvons maintenant construire et utiliser dCAM pour identifier les sous-séquences discriminantes (c’est-à-dire les éventuels précurseurs de vibrations). L’ensemble de l’étape d’entraînement de dCNN et le calcul de dCAM nécessite une journée complète d’exécution sur un GPU NVIDIA V100.

La Figure 6(2) illustre l’application de dCNN et dCAM. Nous allons maintenant mesurer la performance de dCAM pour l’identification de précurseurs de vibrations. Nous divisons l’analyse de la performance en trois parties : (i) temporelle (i.e. analyser à quel moment dCAM permet de détecter/d’anticiper l’événement vibratoire), (ii) structurelle (i.e. identifier les capteurs qui apportent l’information sur l’événement vibratoire), et (iii) forme de sous-séquences (i.e. déterminer des motifs de sous-séquences précurseurs de l’événement vibratoire et leur temporalité). Ces analyses permettent d’évaluer la cohérence des résultats vis-à-vis des connaissances métier.

2) *Identification des précurseurs - analyse temporelle* : Nous mesurons tout d’abord l’évolution du score d’activation obtenu par dCAM. Pour chaque série temporelle multivariée, nous calculons d’abord la moyenne de dCAM pour tous les capteurs (i.e. une série temporelle d’activation pour tous les capteurs). La Figure 8(a) représente les valeurs des quantiles pour chaque index temporel (calculés sur l’ensemble des moyennes de dCAM pour les séries temporelles de \mathbf{T}_M^A). La ligne rouge pleine représente la médiane, tandis que chaque ligne grise en pointillé correspond aux déciles. Nous observons d’abord que le score d’activation moyen est plus élevé lorsque la vibration se produit (ligne verticale rouge dans la Figure 8(a)). Comme il est peu probable de trouver des précurseurs longtemps avant la vibration, nous pouvons ainsi

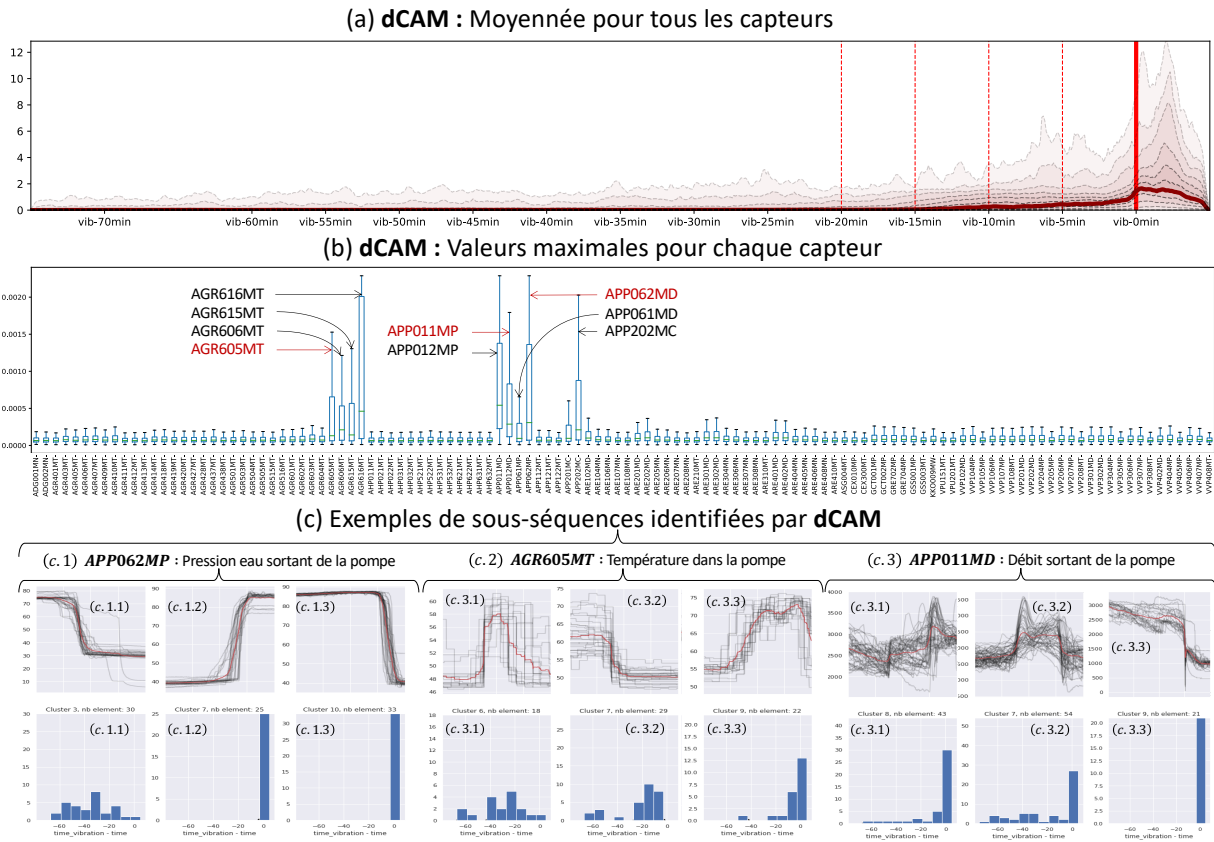


FIGURE 8. Analyse des précurseurs identifiés par dCAM (a) temporellement, (b) structurellement, (c) par sous-séquence.

confirmer que les résultats de dCAM sont cohérents avec la temporalité des crises vibratoires.

3) Identification des précurseurs - analyse structurelle :

Nous mesurons maintenant le score d'activation moyen par capteur. La Figure 8(b) représente la boîte à moustaches du score d'activation de dCAM pour chaque capteur. Nous observons que les scores varient considérablement entre les différents capteurs et nous pouvons facilement en distinguer 9 parmi les 120. Ils correspondent aux mesures de température à l'intérieur des TPA (températures d'étanchéité notées AGR605MT, AGR615MT pour TPA1 et AGR606MT, AGR616MT pour TPA2) et au débit et aux pressions de sortie (notés APP011MD, APP061MP pour TPA1 et APP012MD, APP062MP pour TPA2). Comme le montre la Figure 2(b), AGR et APP sont des sous-systèmes directement connectés aux TPA. De plus, la pression et le débit peuvent influencer directement l'efficacité de la pompe, les zones à faible efficacité étant connues pour être propices aux crises vibratoires. Ainsi, les capteurs mis en évidence par dCAM sont très cohérents avec les connaissances des experts et la structure fonctionnelle de la centrale. De plus, nous pouvons comparer la Figure 8(b) avec la Figure 7(c) et comparer les capteurs activés par dCAM avec les capteurs qui ont fourni la meilleure détection pour NormA et Series2Graph. Nous observons que le même groupe de capteurs est mis en évidence par NormA, Series2Graph et dCAM (AGR et APP), ce qui confirme la cohérence des méthodes non supervisées et supervisées. Cependant, dCAM

sélectionne un plus petit groupe de capteurs, qui est confirmé par l'expert comme étant potentiellement lié à l'apparition des vibrations.

4) Identification des précurseurs - analyse de forme :

Nous analysons enfin les résultats renvoyés par dCAM et discutons des informations que les experts peuvent en tirer. Nous concentrons principalement notre analyse sur 3 des capteurs les plus activés (c'est-à-dire les capteurs en rouge sur la Figure 8(b)). Nous regroupons (avec la méthode de clustering k -means en utilisant la distance euclidienne) les sous-séquences de 15 minutes ayant le score d'activation le plus élevé pour une instance de vibration. La Figure 8(c) représente ces clusters pour les trois capteurs mentionnés ci-dessus. Pour chaque cluster, l'histogramme de distribution temporelle est affiché en dessous. Nous remarquons tout d'abord que la majorité des sous-séquences se sont produites lorsque la vibration a été détectée (comme, par exemple, le groupe représenté sur la Figure 8(c.1.2)). Selon les experts, ces sous-séquences correspondent à une action spécifique (telle qu'une augmentation ou une diminution du débit d'eau dans la TPA1 ou 2 pour augmenter ou diminuer la puissance générée par la centrale), qui pourrait entraîner des vibrations non évitables. Ainsi, dCAM permet d'abord à l'expert de confirmer et de visualiser quelles sous-séquences sont directement corrélées à la vibration. Ensuite, plusieurs sous-séquences détectées par dCAM sont antérieures à la vibration (comme le montrent les Figures 8(c.1.1), (c.3.1), (c.3.2)). Ces sous-séquences pourraient

correspondre à des précurseurs de la vibration et devraient être soigneusement inspectées par les experts. Par exemple, les experts concluent que des clusters tels que ceux de la Figure 8(c.3.1) correspondent à des variations inhabituelles de la température d'étanchéité de la pompe. Globalement, les résultats obtenus sont validés et considérés cohérents et intéressants pour la tâche de détection de précurseurs de vibrations. De plus, l'approche dCNN/dCAM que nous proposons permet aux experts de construire un dictionnaire de motifs liés à une anomalie ciblée. Ce dictionnaire peut être utilisé en fonctionnement pour déclencher des alarmes et aider les experts à éviter d'éventuelles situations critiques.

VI. CONCLUSIONS ET PERSPECTIVES

Dans cet article, nous nous sommes intéressés aux tâches de détection des anomalies et l'identification de leurs précurseurs dans les séries temporelles. Pour la première tâche, nous avons proposé trois nouvelles méthodes (i.e. Series2Graph [4] présenté précédemment, mais aussi NormA [24] et SAND [25] (fonctionnant pour les flux de donnée en temps réel) ainsi que différentes interfaces graphiques utilisateur pour ces méthodes [26], [27]). Pour la deuxième tâche, nous avons de même proposé une nouvelle méthode (i.e. dCNN/dCAM [23] présentée précédemment). Nous avons démontré sur un cas d'application industriel que les méthodes supervisées du type dCNN/dCAM sont significativement plus précises que les méthodes non supervisées, même si NormA et Series2Graph, en plus d'être plus simple d'application et d'utilisation (nécessitant moins de puissance de calcul), fournissent des résultats intéressants et cohérents avec les connaissances des experts. Nous planifions de continuer nos recherches sur le sujet de la détection non-supervisée d'anomalies dans les séries temporelles multivariées. De même, nous étudions la possibilité d'ajouter à dCAM une explication temporelle entre les sous-séquence détectées (i.e. permettant à dCAM de découvrir des chaînes temporelles de patterns pouvant être discriminant et donc précurseurs d'anomalies).

VII. REMERCIEMENT

Ce travail a été financé par EDF R&D et le programme CIFRE de l'ANRT.

RÉFÉRENCES

- [1] A. Bagnall, R. L. Cole, T. Palpanas, and K. Zoumpatianos, "Data Series Management (Dagstuhl Seminar 19282)," *Dagstuhl Reports*, vol. 9, no. 7, pp. 24–39, 2019.
- [2] V. Barnett and T. Lewis, *Outliers in Statistical Data*. John Wiley and Sons, Inc., 1994.
- [3] C.-C. M. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, D. F. Silva, A. Mueen, and E. Keogh, "Matrix profile i : All pairs similarity joins for time series : A unifying view that includes motifs, discords and shapelets," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, 2016, pp. 1317–1322.
- [4] P. Boniol and T. Palpanas, "Series2graph : Graph-based subsequence anomaly detection for time series," *Proc. VLDB Endow.*, vol. 13, no. 12, p. 1821–1834, Jul. 2020.
- [5] D. Hawkins, *Identification of Outliers*. Springer Netherlands, 1980.
- [6] M. Hadjem, F. Naït-Abdesselam, and A. Khokhar, "St-segment and t-wave anomalies prediction in an ecg data using rusboost," in *2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom)*, 2016, pp. 1–6.
- [7] P. Senin, J. Lin, X. Wang, T. Oates, S. Gandhi, A. P. Boedihardjo, C. Chen, and S. Frankenstein, "Time series anomaly discovery with grammar-based compression," in *EDBT*, 2015.
- [8] J. Antoni and P. Borghesani, "A statistical methodology for the design of condition indicators," *Mechanical Systems and Signal Processing*, vol. 114, pp. 290–327, 2019.
- [9] D. Yankov, E. Keogh, and U. Rebbapragada, "Disk aware discord discovery : Finding unusual time series in terabyte sized datasets," in *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, 2007, pp. 381–390.
- [10] Q. Yang and X. Wu, "10 challenging problems in data mining research," *International Journal of Information Technology and Decision Making (IJITDM)*, vol. 05, no. 04, pp. 597–604, 2006.
- [11] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, "The great time series classification bake off : a review and experimental evaluation of recent algorithmic advances," *Data Mining and Knowledge Discovery*, vol. 31, p. 606–660, 11 2017.
- [12] H. A. Dau, A. J. Bagnall, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. Ratanamahatana, and E. J. Keogh, "The ucr time series archive," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, pp. 1293–1305, 2019.
- [13] A. Bagnall, J. Lines, J. Hills, and A. Bostrom, "Time-series classification with cote : The collective of transformation-based ensembles," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 9, pp. 2522–2535, 2015.
- [14] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification : a review," *Data Mining and Knowledge Discovery*, vol. 33, pp. 917–963, 2019.
- [15] Z. Cui, W. Chen, and Y. Chen, "Multi-scale convolutional neural networks for time series classification," *CoRR*, vol. abs/1603.06995, 2016.
- [16] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, "Time series classification using multi-channels deep convolutional neural networks," in *WAIM*, 2014.
- [17] B. Zhao, H. zhang Lu, S. Chen, J. Liu, and D. Wu, "Convolutional neural networks for time series classification," *Journal of Systems Engineering and Electronics*, vol. 28, pp. 162–169, 2017.
- [18] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks : A strong baseline," *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 1578–1585, 2017.
- [19] H. Ismail Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. Schmidt, J. Weber, G. Webb, L. Idoumghar, P. Muller, and F. Petitjean, "Inceptiontime : finding alexnet for time series classification," *Data Mining and Knowledge Discovery*, vol. 34, p. 1936–1962, Sep. 2020.
- [20] D. P. Kingma and J. Ba, "Adam : A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.
- [21] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Evaluating surgical skills from kinematic data using convolutional neural networks," in *MICCAI*, 2018.
- [22] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2921–2929, 2016.
- [23] P. Boniol, M. Mefteh, E. Remy, and T. Palpanas, "dcam : Dimension-wise class activation map for explaining multivariate data series classification," *Proceedings of the 2022 International Conference on Management of Data (SIGMOD'22)*, 2022.
- [24] P. Boniol, M. Linardi, F. Roncallo, T. Palpanas, M. Mefteh, and E. Remy, "Unsupervised and scalable subsequence anomaly detection in large data series," *The VLDB Journal*, 2021.
- [25] P. Boniol, J. Paparrizos, T. Palpanas, and M. J. Franklin, "SAND : streaming subsequence anomaly detection," *Proc. VLDB Endow.*, vol. 14, no. 10, pp. 1717–1729, 2021.
- [26] P. Boniol, T. Palpanas, M. Mefteh, and E. Remy, "Graphan : Graph-based subsequence anomaly detection," *Proc. VLDB Endow.*, vol. 13, no. 12, p. 2941–2944, Aug. 2020.
- [27] P. Boniol, M. Linardi, F. Roncallo, and T. Palpanas, "Sad : An unsupervised system for subsequence anomaly detection," in *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, 2020, pp. 1778–1781.