Approximation du score CFOF de détection d'anomalie dans un arbre d'indexation iSAX : Application au contexte SI de la SNCF

Lucas Foulon, Christophe Rigotti, Serge Fenet, Denis Jouvin

Approximation du score CFOF de détection d'anomalie dans un arbre d'indexation iSAX : Application au contexte SI de la SNCF

Lucas Foulon[1], Christophe Rigotti[2], Serge Fenet[3], and Denis Jouvin[4]

[1] Univ Lyon, CNRS, LIRIS, UMR5205, F-69621, Villeurbanne, France
et SNCF Mobilité, DSI Production Ferroviaire, F-69393, Lyon, France
lucas.foulon@sncf.fr
[2] Univ Lyon, INSA-Lyon, CNRS, INRIA,
LIRIS, UMR5205, F-69621, Villeurbanne, France
christophe.rigotti@insa-lyon.fr
[3] Univ Lyon, Université Claude Bernard Lyon 1, CNRS,
LIRIS, UMR5205, F-69621, Villeurbanne, France
serge.fenet@liris.cnrs.fr
[4] SNCF Mobilité, DSI Production Ferroviaire, F-69393, Lyon, France
denis.jouvin@sncf.fr

Abstract. La finalité de notre travail est la détection des anomalies dans les traces de fonctionnement de l'infrastructure de communication du Système d'Information (SI) de la SNCF. Deux techniques récentes et indépendantes semblent particulièrement appropriées dans notre cas. Il s'agit d'une part du stockage et de l'indexation de séries temporelles dans un arbre appelé arbre iSAX, et d'autre part d'un score de détection d'anomalie nommé CFOF dont la robustesse au phénomène de concentration en haute dimension a été établie de façon formelle. Dans cet article nous montrons qu'il est possible d'utiliser la structuration des informations dans l'arbre iSAX pour déterminer rapidement une approximation du score CFOF. La valeur obtenue est proche du score exact sur des données synthétiques et réelles. Les premiers retours d'expertises indiquent que la méthode semble pertinente pour le déclenchement d'alarmes sur les données issues de trace d'activité du SI de la SNCF.

1 Introduction

Une anomalie peut être définie comme une déviation par rapport à ce qui est défini comme normal. La détection d'anomalie constitue une tâche importante dans de nombreux domaines tels que l'analyse de données, la reconnaissance d'image médicale, la détection d'intrusion dans les systèmes informatiques, ou encore la fraude à la carte bancaire. Avec le développement croissant des volumes de données issues des applications métier, la détection d'anomalie représente par ailleurs un enjeu de plus en plus important dans de nombreux domaines industriels.

Dans le contexte industriel de la SNCF[5], l'objectif de ce travail est de détecter un comportement anormal dans les traces de messages au sein de son système d'information (SI). L'historique des séries temporelles observées par le passé est indexé à l'aide d'un arbre iSAX pour permettre son archivage et son interrogation. En effet, les arbres iSAX [10], [11] sont des structures d'indexation multidimensionnelles très performantes pour les séries temporelles. Elles permettent des recherches par similarité très efficaces sur des critères de distance, mais sont aussi parmi les seules à supporter la pondération des portions de signaux et la déformation temporelle dynamique (*dynamic time warping*) durant la recherche. Elles ont également montré qu'elles restaient opérationnelles même lorsque la base de séries temporelles dépasse le milliard de séries.

Sur ces données, nous nous intéressons à la détermination du score de détection d'anomalie CFOF proposé récemment dans [3]. L'intérêt particulier de ce score est qu'il soit actuellement le seul pour lequel la robustesse vis-à-vis de l'augmentation de la dimensionnalité des données ait été établie de façon formelle et expérimentale. Notre contribution principale est de montrer qu'il est possible de tirer parti des propriétés des arbres iSAX pour calculer de façon efficace une approximation du score CFOF. La méthode proposée a été testée sur des jeux de données synthétiques, ainsi que sur des données SNCF. Dans tous les cas, les scores approchés sont très proches des scores réels. Enfin, l'interprétation des résultats sur les données réelles confirme la pertinence de l'utilisation de la méthode.

2 Positionnement par rapport aux approches existantes

La détection d'anomalies est toujours un domaine très actif obtenant de nombreux résultats, comme par exemple les travaux récents présentés dans [1] pour aider au diagnostic de la rétinopathie diabétique grâce à l'utilisation d'un réseau de neurones convolutionnel. Il existe de multiples méthodes appliquées à des domaines très différents [6], [5]. La partie 2.1 suivante décrit quelques unes de ces méthodes.

2.1 Différentes méthodes

On peut trouver parmi ces méthodes des approches non supervisées telle que l'utilisation de forêt d'arbres d'isolation présentée dans [12]. Ces arbres d'isolation travaillent chacun sur un échantillon des données, et les données marquées comme *isolées* par plusieurs arbres sont considérées comme *anormales*. Il existe aussi des méthodes supervisées, comme par exemple [9], qui utilise un classifieur de type *Support Vector Machine* dans le cadre de détection d'intrusions système (programme américain Defense Advanced Research Projects Agency 1998). Parmi ces méthodes supervisées, on notera également l'utilisation de règles et de motifs, comme notamment dans la technique proposée dans [8] pour détecter des trajectoires anormales d'objets.

[5] Société Nationale des Chemins de fer Français.

2.2 Méthodes basées sur une notion de proximité

Ces méthodes sont largement utilisées, avec de nombreuses variantes. On peut y trouver trois sous-familles [2] : les méthodes basées sur du clustering, celles basées sur des distances et celles basées sur des densités.

Les méthodes basées sur du clustering observent si un objet appartient ou non à un cluster. L'objet est considéré anormal s'il ne se trouve rattaché à aucun cluster [9]. Les méthodes basées sur des distances les plus typiques utilisent simplement, pour calculer le score d'anomalie d'un objet q, les distances entre q et ses k plus proches voisins [7]. La troisième sous-famille, basée sur des densités, contient des techniques qui vont tenir compte des objets dans une zone *proche* autour de l'objet q à évaluer. Ces objets et leur distribution influenceront le score attribué à q [4].

Une des meilleures méthodes de détection par densité est celle basée sur le *Local Outlier Factor* introduite dans [4]. Cependant, sa pertinence diminue très fortement lorsque la dimensionnalité de l'espace augmente, à cause du phénomène communément appelé *malédiction de la dimensionnalité*. En effet, lorsque la dimension augmente, les valeurs des distances entre objets tendent à être plus similaires (alors que les objets, eux, ne le sont pas forcément) et les méthodes sont confrontées au problème dit de *concentration* des objets. Comme montré dans [3], le score *Local Outlier Factor* n'échappe pas à cela, et il tend vers 1 pour tous les objets lorsque la dimensionnalité augmente. C'est pourquoi [3] propose une nouvelle méthode appelé CFOF (*Concentration Free Outlier Factor*) ayant la propriété de résister à ce phénomène de concentration. Cette méthode sera plus amplement détaillée dans la section 3.1.

En complément, [3] propose une technique efficace de calcul de CFOF par échantillonnage, où les scores CFOF de tous les objets d'un échantillon sont calculés par rapport à tous les autres objets du même échantillon. Cette technique tire partie d'une factorisation des opérations nécessaires au sein de chaque échantillon. La qualité des estimations dépend de la taille de l'échantillon, et si cette technique est bien adaptée lorsque l'on souhaite calculer les scores pour tous les objets d'une base, elle ne l'est pas lorsque l'on veut calculer seulement le score d'un nouvel objet vis-à-vis d'un historique de référence.

Nous nous plaçons dans le cadre d'un stockage existant d'un historique de séries temporelles dans un arbre iSAX [10], [11], qui est une structure d'indexation particulièrement performante pour l'interrogation et la recherche par similarité. La méthode que nous présentons permet de calculer une approximation du score CFOF d'une nouvelle série en tirant parti des propriétés de l'arbre iSAX.

3 Proposition

La méthode introduite dans cet article permet, lorsque des séries temporelles sont stockées et indexées dans un arbre iSAX [10], [11], de tirer parti des propriétés de cet arbre pour calculer le score d'anomalie CFOF [3] d'une nouvelle série. Nous allons tout d'abord rappeler brièvement la définition du score CFOF et nous présenterons ensuite la méthode proposée.

3.1 Rappel de la définition du score CFOF

Le calcul du score CFOF [3] d'un objet q vis-à-vis d'un ensemble d'objets de référence $\mathcal{R}$ consiste à chercher la taille du voisinage minimale k_m telle que q soit dans les k_m plus proches voisins d'au moins une fraction ϱ des objets de $\mathcal{R}$. Le score CFOF(q) est alors la valeur de k_m normalisée par rapport à la taille de $\mathcal{R}$. Ce score est paramétré par le seuil ϱ (dans $[0;1]$) qui détermine la proportion d'objets de $\mathcal{R}$ devant inclure q dans leur voisinage (de taille k_m). Ce paramètre rend la valeur du score plus ou moins sensible au nombre d'objets de référence auxquels q doit ressembler, et comme le montrent les expériences de la section 4 son réglage est simple en pratique.

De façon plus formelle, la mesure se définit comme suit. Soit $\mathrm{nn}_k(x)$ le $k$ième plus proche voisin d'un objet x, c'est-à-dire tel qu'il existe seulement $k-1$ objets plus proches de x que ne l'est l'objet $\mathrm{nn}_k(x)$. L'ensemble des k plus proches voisins de x est noté $\mathrm{NN}_k(x)$ et inclut tous les objets tel que $\{\mathrm{nn}_i(x) \mid 1 \leq i \leq k\}$. Soit $\mathrm{N}_k(x)$ le nombre d'objets de référence ayant x parmi leur k plus proches voisins et défini par $\mathrm{N}_k(x) = |\{y \mid x \in \mathrm{NN}_k(y)\}|$. Le score CFOF d'un objet q est alors : $\mathrm{CFOF}(q) = \min\{k/|\mathcal{R}| : \mathrm{N}_k(q) \geq \varrho \times |\mathcal{R}|\}$

Le score CFOF de détection d'anomalies est actuellement le seul pour lequel il a été montré de façon formelle et constaté de façon expérimentale [3] qu'il n'était pas sensible au phénomène de concentration quand la dimensionnalité des données augmente.

3.2 Principe général de la méthode de calcul proposée

Dans notre contexte, l'historique de référence (par exemple une mesure au fil du temps) est découpé en séries de longueur fixe pouvant se chevaucher ou non. Une série de longueur $\mathcal{D}$ (contenant $\mathcal{D}$ valeurs) est alors considérée comme étant un objet dans un espace à $\mathcal{D}$ dimensions, et chacune des séries est stockée sous la forme d'un objet indexé dans l'arbre iSAX.

Pour un nouvel objet q (une nouvelle série), qui n'est pas dans l'arbre iSAX, la méthode que nous proposons permet de calculer une approximation du score CFOF de q par rapport aux objets de référence stockés dans l'arbre iSAX, en tirant parti des propriétés de cet arbre.

Dans les arbres iSAX, comme dans tout arbre d'indexation en général, une feuille contient un ensemble d'objets similaires, mais un arbre iSAX possède deux autres propriétés dont nous allons tirer parti :

1. Les feuilles ne se chevauchent pas, et une zone de l'espace multidimensionnel n'est représentée que par une feuille.
2. L'indexation utilisée permet de borner les distances lors des recherches dans l'arbre. Par exemple, pour un objet p (contenu dans l'arbre ou non), il est possible au niveau de tout nœud $\mathcal{N}$ (intermédiaire ou feuille) de connaître une borne inférieure de la distance entre p et l'objet le plus proche de p indexé dans le sous-arbre à partir de $\mathcal{N}$.

La première propriété est importante car elle va permettre de précalculer et de stocker des statistiques liées à la distribution des objets pour tout l'espace couvert par l'arbre, et ce à une granularité qui est celle de la feuille. La seconde propriété sera quant à elle mise à profit pour élaguer, lors du calcul d'un score CFOF, les zones de l'espace ne pouvant pas contenir d'objets participant au voisinage en cours de détermination.

Pour calculer le score CFOF d'un nouvel objet q par rapport aux objets de référence stockés dans l'arbre iSAX, une des principales difficultés est d'arriver à obtenir pour chacun des objets p stockés dans l'arbre la valeur du rang de q dans le voisinage de p. Ce rang vaudra 1 si q est le plus proche voisin de p, 2 si q est le second plus proche voisin de p, etc. La valeur de ce rang, notée $v\text{-}rang_p(q)$, est définie plus précisément comme étant la valeur de k pour laquelle $\mathrm{nn}_k(p) = q$. Lorsque ces valeurs $v\text{-}rang_p(q)$ sont connues pour tous les objets p de référence, l'obtention de CFOF(q) est simple. Il suffit de placer ces valeurs dans une liste triée par ordre croissant, que nous noterons $l_{v\text{-}rang}$, et de prendre dans $l_{v\text{-}rang}$ l'élément d'indice $\lceil \varrho \times |\mathcal{R}| \rceil$ (obtenu par arrondi supérieur). La valeur de cet élément correspond à la taille de voisinage minimale k_m telle que q soit dans les k_m plus proches voisins d'au moins une fraction ϱ des objets de $\mathcal{R}$. La valeur de CFOF(q) est alors la valeur normalisée de k_m par rapport à la taille de $\mathcal{R}$, c'est-à-dire $l_{v\text{-}rang}[\lceil \varrho \times |\mathcal{R}| \rceil]/|\mathcal{R}|$.

Revenons sur l'étape clef de calcul de $v\text{-}rang_p(q)$. Pour determiner cette valeur nous devons obtenir le nombre d'objets de référence r tels que $distance(p, r) \leq distance(p, q)$. Afin d'éviter d'avoir à compter chaque objet r un par un, nous proposons de calculer une approximation de $v\text{-}rang_p(q)$ en utilisant des fonctions de répartition des distances des objets dans différentes zones de l'espace. Le calcul de $v\text{-}rang_p(q)$ s'effectue alors en parcourant toutes les zones non vides de l'espace, c'est-à-dire toutes les feuilles $\mathcal{N}_f$ de l'arbre iSAX, et en utilisant une fonction de répartition pour déterminer le nombre d'objets r de $\mathcal{N}_f$ qui sont tels que $distance(p, r) \leq distance(p, q)$. Pour des vecteurs dont les composantes suivent une loi normale, le module de ces vecteurs suit une loi du χ (apparentée à la loi du χ^2). C'est une généralisation d'autres lois, notamment de la loi de Maxwell qui décrit la distribution des modules de vitesses de particules en 3 dimensions. Un aspect intéressant de cette loi du χ est qu'elle tend rapidement vers une loi normale quand la dimensionnalité augmente. Notons $F_{\mu,\sigma}(x)$ la fonction de répartion de la loi normale de moyenne μ et d'écart type σ. Soit, pour une feuille $\mathcal{N}_f$ et un objet p, des approximations $\tilde{\mu}$ et $\tilde{\sigma}$ de la moyenne et de l'écart type des distances entre p et les objets contenus dans $\mathcal{N}_f$. La fraction d'objets de $\mathcal{N}_f$ situés à une distance de p inférieure ou égale à $distance(p, q)$ peut alors être approximée par $F_{\tilde{\mu},\tilde{\sigma}}(distance(p, q))$. C'est ce principe qu'utilise l'algorithme détaillé dans la section suivante.

3.3 Algorithme d'approximation de CFOF à partir d'un arbre iSAX

Le pré-traitement nécessaire à l'exécution de l'algorithme est le calcul des approximations $\tilde{\mu}$ et $\tilde{\sigma}$ pour chaque objet de référence p vis-à-vis de chaque nœud feuille $\mathcal{N}_f$. Pour un objet p et un nœud feuille $\mathcal{N}_f$, $\tilde{\mu}$ est noté $\widetilde{\texttt{dist}}(\mathcal{N}, p)$ et

$\tilde{\sigma}$ est noté $\tilde{\sigma}_{\mathcal{N}}(p)$. Le principe de l'algorithme proposé est indépendant de ces approximations, et nous détaillerons leurs calculs ensuite dans la Section 3.4.

En plus des valeurs $\tilde{\mu}$ et $\tilde{\sigma}$, l'approximation du score CFOF d'un objet q à partir des objets de référence contenus dans un arbre iSAX, nécessite d'autres paramètres d'entrée : la racine $\mathcal{N}_{racine}$ de l'arbre ISAX, l'ensemble $\mathcal{R}$ des objets de référence (en pratique cet ensemble peut aussi être obtenu à partir de l'arbre) et le paramètre ϱ. L'algorithme 1 décrit le calcul de l'estimation CFOF q à partir de ces entrées.

Le principe général est celui présenté dans la section précédente. Pour chaque objet de référence p l'algorithme calcule $v\text{-}rang_p(q)$ par cumul dans la variable *v-rang* (boucle commençant à la ligne 2). Chaque valeur de *v-rang* est insérée dans la liste triée $l_{v\text{-}rang}$, permettant de retourner la valeur de l'approximation de $CFOF(q)$ (ligne 24).

Pour le calcul de $v\text{-}rang_p(q)$, pour un p donné dans la boucle principale, l'algorithme parcourt l'arbre à partir de sa racine (boucle interne commençant ligne 6) en tenant à jour une liste $liste_{\mathcal{N}}$ des nœuds restants à visiter. Le nœud courant est ôté de cette liste ligne 7. Ensuite l'algorithme utilise les bornes `minDist` et `maxDist` sur la distance entre p et les objets contenus dans le sous-arbre associé au nœud courant, qui sont des bornes fournies par la structure d'indexation iSAX. Deux élagages sont alors réalisés :
1. Si la distance minimale entre p et les objets représentés par le nœud courant est supérieure à la distance *dist* entre p et q (ligne 8), alors le sous-arbre ne contient pas d'objet à compter dans $v\text{-}rang_p(q)$. L'exploration du sous-arbre peut alors être évitée de façon certaine.
2. Si c'est la distance maximale qui est inférieure à la distance *dist* (ligne 10), alors tous les objets du sous-arbre sont plus proches de p que ne l'est q. Il est donc possible de réaliser un second type d'élagage sûr, en comptant tous ces objets directement dans $v\text{-}rang_p(q)$ sans parcourir le sous-arbre. Ceci est réalisé ligne 11 avec $nbrObj(\mathcal{N}_{courant})$ représentant le nombre d'objets du sous-arbre du nœud $\mathcal{N}_{courant}$.

S'il n'y a pas eu d'élagage, deux cas sont possibles selon que $\mathcal{N}_{courant}$ soit un nœud feuille ou pas. Si c'est un nœud feuille (ligne 12), alors l'algorithme va comptabiliser dans $v\text{-}rang_p(q)$ les objets du nœud qui sont plus proches de p que ne l'est q, en approximant ce décompte avec la fonction de répartition $F_{\tilde{\mu},\tilde{\sigma}}$. Enfin, si le nœud courant est un nœud interne de l'arbre sans possibilité d'élagage (ligne 16), alors les nœuds enfants immédiats de $\mathcal{N}_{courant}$ dans l'arbre sont ajoutés à la liste des nœuds restants à visiter.

3.4 Calcul des paramètres $\widetilde{\texttt{dist}}(\mathcal{N}, p)$ et $\tilde{\sigma}_{\mathcal{N}}(p)$

Même si le principe de l'algorithme et des élagages présentés Section 3.3 sont indépendants des paramètres $\tilde{\mu}$ et $\tilde{\sigma}$ de la fonction de répartition F, ces paramètres vont influer sur la qualité globale de l'approximation de CFOF réalisée. Nous indiquons ici les valeurs utilisées dans les expériences présentées Section 4, et qui ont permis d'obtenir une très bonne approximation du score CFOF tant sur

Algorithme 1 : Calcul de l'approximation du score CFOF de q dans un arbre iSAX

Data : L'objet q, l'ensemble $\mathcal{R}$ des objets de référence, la racine $\mathcal{N}_{racine}$ de l'arbre, les valeurs $\widetilde{\texttt{dist}}(\mathcal{N}_f, p)$ et $\tilde{\sigma}_{\mathcal{N}_f}(p)$, le paramètre CFOF ϱ

$l_{v\text{-}rang} \leftarrow \emptyset$
forall p *dans* $\mathcal{R}$ **do**
  $v\text{-}rang \leftarrow 0$
  $dist \leftarrow \texttt{distance}(p, q)$
  $liste_{\mathcal{N}} \leftarrow [\mathcal{N}_{racine}]$; `// nœuds restants à parcourir`
  while $liste_{\mathcal{N}} \neq \emptyset$ **do**
    $\mathcal{N}_{courant} \leftarrow liste_{\mathcal{N}}.pop()$
    if $\texttt{minDist}(p, \mathcal{N}_{courant}) \geq dist$ **then**
      Rien à faire, ignorer simplement $\mathcal{N}_{courant}$
    else if $\texttt{maxDist}(p, \mathcal{N}_{courant}) \leq dist$ **then**
      $v\text{-}rang \leftarrow v\text{-}rang + nbrObj(\mathcal{N}_{courant})$
    else if $\mathcal{N}_{courant}$ *est un nœud feuille* **then**
      $\tilde{\mu} \leftarrow \widetilde{\texttt{dist}}(\mathcal{N}_{courant}, p)$
      $\tilde{\sigma} \leftarrow \tilde{\sigma}_{\mathcal{N}_{courant}}(p)$
      $v\text{-}rang \leftarrow v\text{-}rang + F_{\tilde{\mu},\tilde{\sigma}}(dist) * nbrObj(\mathcal{N}_{courant})$
    else
      forall $\mathcal{N}$ *dans* $\mathcal{N}_{courant}.enfants$ **do**
        Insérer $\mathcal{N}$ dans $liste_{\mathcal{N}}$
      end
    end
  end
  Insérer $v\text{-}rang$ dans $l_{v\text{-}rang}$ par ordre croissant
end
return $l_{v\text{-}rang}[\lceil \varrho \times |\mathcal{R}| \rceil]/|\mathcal{R}|$

des données synthétiques de distributions gaussiennes que sur les données réelles traitées dans le système d'information de la SNCF.

Tout d'abord pour $\tilde{\mu}$, c'est-à-dire $\widetilde{\texttt{dist}}(\mathcal{N}, p)$ pour un objet p et un nœud $\mathcal{N}$, c'est la moyenne quadratique aussi appelée RMS (*Root Mean Square*) qui est utilisée :

$$\widetilde{\texttt{dist}}(\mathcal{N}, p) = \sqrt{\frac{1}{|\mathcal{N}|} \times (\sum_{r \in \mathcal{N}} |r - p|^2)}$$

avec $\mathcal{N}$ dénotant ici l'ensemble des objets contenus dans le nœud lui-même. Soit C le barycentre des objets de $\mathcal{N}$ (en prenant une masse unitaire pour chaque objet). Par le théorème de *Huygens* nous avons :

$$\sum_{r \in \mathcal{N}} |r - p|^2 = \sum_{r \in \mathcal{N}} |r - C|^2 + |\mathcal{N}| \times |C - p|^2$$

c'est-à-dire :

$$\widetilde{\texttt{dist}}(\mathcal{N}, p) = \sqrt{\frac{1}{|\mathcal{N}|} \times (\sum_{r \in \mathcal{N}} |r - C|^2 + |\mathcal{N}| \times |C - p|^2)}$$

Nous pouvons donc pré-calculer $\sum_{r \in \mathcal{N}} |r - C|^2$ pour chaque feuille de l'arbre iSAX et ensuite déterminer simplement $|C - p|^2$ lorsqu'il est nécessaire d'obtenir la valeur de $\widetilde{\texttt{dist}}(\mathcal{N}, p)$.

L'autre étape de pré-traitement est le calcul des écarts types $\tilde{\sigma}$, c'est-à-dire des valeurs $\tilde{\sigma}_{\mathcal{N}}(p)$. Pour cela, nous utilisons une moyenne pondérée des écarts types pris sur chacune des dimensions, afin de privilégier les dimensions où p s'écarte le plus de C (le barycentre des objets du nœud $\mathcal{N}$). Soit $\mathcal{D}$ le nombre de dimensions et $\sigma_1, \sigma_2, \ldots, \sigma_{\mathcal{D}}$ les écarts types des objets de $\mathcal{N}$ pour chaque dimension. Soit $(p_1, p_2, \ldots, p_{\mathcal{D}})$ les coordonnées de p et $(C_1, C_2, \ldots, C_{\mathcal{D}})$ celle de C. La valeur de $\tilde{\sigma}_{\mathcal{N}}(p)$ est alors :

$$\tilde{\sigma}_{\mathcal{N}}(p) = \sum_{d=1}^{\mathcal{D}} (\sigma_d * \frac{|C_d - p_d|}{\sum_{i=1}^{\mathcal{D}} |C_i - p_i|})$$

4 Évaluations et analyses

Nous avons évalué notre approche sur deux ensembles de tests. Le premier, construit sur le même jeu de données que celui utilisé dans [3], est destiné à estimer la capacité de notre algorithme à effectuer une approximation correcte du score CFOF. Le second utilise un jeu de données issu du SI de la SNCF, et sert à évaluer sa capacité à détecter des anomalies réelles identifiées et qualifiées par les experts.

4.1 Évaluation sur le jeu de données *Clust2*

La première évaluation a pour but de vérifier la qualité de notre estimation du score CFOF. Pour ce faire, nous utilisons le jeu de données *Clust2* de [3], généré de la manière suivante : pour un ensemble de dimensions $\mathcal{D} \in [2, 5, 10, 20, 50]$, 10000 points sont générés selon deux distributions normales. La première est centrée sur l'origine avec un écart-type de 1 sur chaque dimension, et la seconde est centrée sur $(4 \ldots 4)$ avec un écart-type de 0.5. Dans cette première évaluation, le paramètre CFOF ϱ est fixé à 0.1. Pour chaque point, nous calculons d'une part le score CFOF réel, et d'autre part son approximation avec l'algorithme 1.

Les résultats de cette évaluation sont présentés sur la figure 1. Elle montre une comparaison du score exact (en bleu) et du score estimé (en jaune) pour les jeux de données en 2, 5, 10, 20 et 50 dimensions, par ordre croissant de score CFOF exact (et donc par ordre d'anormalité croissante). Nous voyons que quelle que soit la dimension des données, la courbe d'estimation suit la courbe du score

exacte avec beaucoup de précision. Nous voyons aussi qu'au fur et à mesure que l'on augmente le nombre de dimensions, une faible transition a lieu : jusqu'en dimension 10 le score estimé est légèrement inférieur au score réel pour toutes les valeurs du score. À partir de la dimension 20, le score devient légèrement surestimé pour les faibles valeurs de score (pour les 2000 premiers points). Cette tendance s'accentue en 50 dimensions, où le score est surestimé pour les 6000 premiers points.

La figure en 20 dimensions représente par ailleurs, une estimation de l'apport lié à l'utilisation de l'arbre iSAX. En effet, la structure hiérarchique de l'arbre reflète l'organisation multi-dimensionnelle du nuage des points qui y sont stockés, et notamment l'anisotropie de sa distribution spatiale. Nous pouvons évaluer l'apport de cette structure en faisant la supposition que pour chaque nœud, les objets stockés sont distribués uniformément entre les distances minimale et maximales aux objets de ce nœud. Nous pouvons calculer un score CFOF avec cette estimation, en nous attendant à ce que ce score soit de plus mauvaise qualité que celui obtenu en exploitant la structure de l'arbre. C'est ce que l'on constate sur la courbe en 20 dimensions, avec le score tracé en vert. Là aussi, nous voyons que le score est surestimé pour les 2000 premiers objets, et sous-estimé pour les objets suivants, mais avec une différence bien plus grande par rapport à notre algorithme. Le dernier graphique de la figure 1 présente un zoom sur les 500 derniers scores en 20 dimensions, montrant que l'apport de la structure arborescente iSAX est indispensable et que sans elle l'approximation du score CFOF n'est pas exploitable.

Concernant la rédution du temps d'exécution, sur le jeu en 20 dimensions, le temps moyen de calcul de notre approximation du score CFOF est de 40 secondes par objet (machine Linux équipée d'un Intel Xeon Silver 4114 à 2.2 GHz), et de plus de 40 fois cette valeur pour le calcul du score exact.

4.2 Évaluation sur données réelles de la SNCF

La seconde évaluation de notre algorithme porte sur un jeu de données réel issu du contexte industriel de la plateforme de médiation SNCF appelée CanalTrain. Cette évaluation sert à mettre en avant trois points importants dans la cadre de l'application industrielle. Tout d'abord, nous comparons le score CFOF estimé et le score réel sur les vraies données métier, ce qui nous permet de constater que notre estimation est de qualité suffisante pour l'application industrielle envisagée. Ensuite, l'interprétation des résultats montre une utilisation possible dans le cadre de flux de messages, où les nouvelles séries sont construites par agrégation des données arrivant, et leurs scores CFOF sont calculés par rapport à l'arbre iSAX contenant les séries relatives à un historique donné. Enfin, plusieurs scores sont calculés en faisant varier le paramètre ϱ. Nous montrons que ce paramètre permet de régler la sensibilité de la détection d'anomalie : avec un paramètre ϱ faible, seules les anomalies les plus grossières sont identifiées. Avec un paramètre plus élevé, les anomalies plus fines sont aussi identifiées et donnent lieu à un score CFOF plus élevé. Dans notre application, le réglage de ce paramètre se fera en

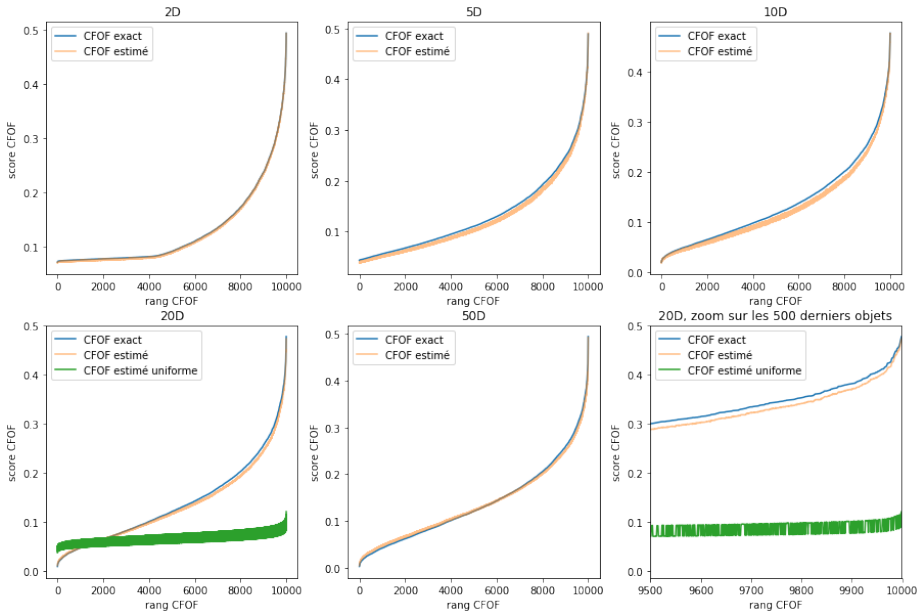

Fig. 1. Les cinq premiers graphiques comparent le score CFOF réel et le score CFOF estimé, pour des objets de 2, 5, 10, 20 et 50 dimensions. Le graphique en 20 dimensions illustre l'évaluation de la distance proportionnelle. Le dernier graphique présente un zoom sur les 500 derniers scores.

fonction du taux de faux négatifs levés et des retours des experts, mais nous montrons d'ores et déjà qu'un paramètre unique permet de piloter la détection.

L'historique des données d'apprentissage couvre la période de novembre 2017 à fin août 2018, et porte sur le nombre de messages par minute sur la plateforme de médiation. Dans une première phase de pré-traitement, nous générons à partir des données brutes un ensemble de séries de mesures. Chaque série comporte 24 mesures (c'est un objet en 24 dimensions) et couvre une fenêtre temporelle de 6 heures. Chacune de ces 24 mesures représente le nombre de messages reçus sur 15 minutes. Les séries se recouvrent dans le temps, et le début d'une série est décalé de 30 minutes par rapport au début de la précédente. Nous obtenons ainsi 13883 séries, chacune contenant 24 valeurs nous informant sur le nombre de messages reçus par pas de temps de 15 minutes. Ces séries sont ensuite insérées dans l'arbre, qui constitue la base de référence à laquelle nous allons comparer le reste des données.

La même méthode de d'extraction à partir des donnée brutes de séries temporelles est appliquée aux données disponibles sur la période du 1er au 24 septembre 2018, ce qui nous donne 1122 séries. Nous estimons ensuite le score CFOF de chacune de ces séries par rapport aux séries de référence stockées dans l'arbre. À des fins d'évaluation et de comparaison, nous calculons aussi leur score exact,

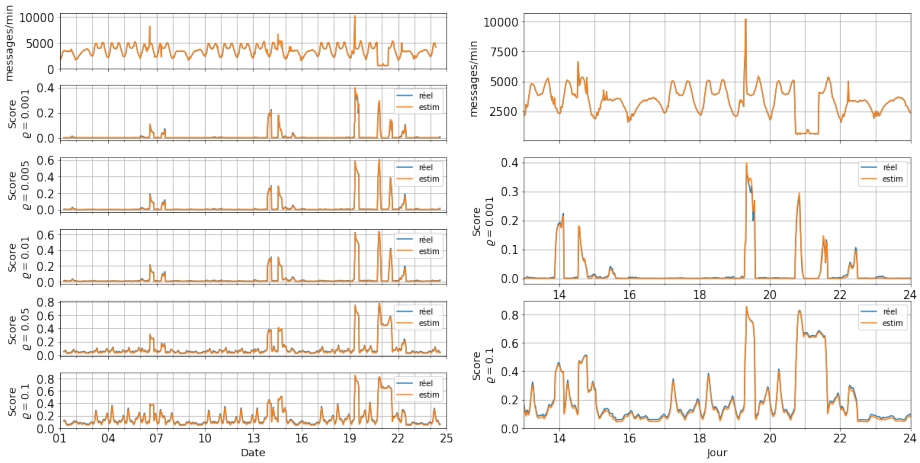

Fig. 2. À gauche : la première courbe représente le nombre de message par minutes. Les cinq courbes suivantes représentent le score CFOF avec $\varrho = (0.001, 0.005, 0.01, 0.05, 0.1)$. À droite : zoom sur la période du 13 au 23 septembre, avec $\varrho = (0.001, 0.1)$.

mais dans l'application finale cette phase sera omise. Nous voyons que les deux courbes correspondantes, présentées sur la figure 2, sont presque tout le temps confondues. Plusieurs valeurs du paramètre ϱ sont évaluées, afin de voir dans quelle mesure nous pouvons contrôler le taux de faux négatifs.

Pour ce jeu de données, les exécutions ont également été réalisées sur un Intel Xeon Silver 4114 à 2.2 GHz, et se terminent avec un temps moyen de calcul de notre approximation du score CFOF de 160 secondes par objet, alors que le temps mesuré pour le calcul du score exact est plus de 25 fois supérieur.

Toutes les anomalies détectées avec un paramètre ϱ de 0.001 correspondent à des incidents métiers réels. Nous en discutons trois en particulier ici. Une première anomalie, identifiée autour du 7 septembre, provient de l'accumulation de messages provenant d'une plateforme en amont. Notre méthode permet de détecter d'une part un pic anormal du nombre de messages entre le 6 et le 7 septembre, et d'autre part une baisse anormale le matin du 7.Une deuxième anomalie, représentative des anomalies souvent détectées, est visible le 19 septembre au matin. Nous voyons que le nombre de messages a arrêté d'augmenter de 3h à 6h du matin, avant de brusquement grimper à plus de 10000 messages par minute. C'est un cas d'anomalie assez classique dans lequel la plateforme n'arrive pas à traiter tous les messages arrivant suite à une saturation, lorsqu'une ou plusieurs files d'attente en amont de CanalTrain vident massivement leurs buffers de messages accumulés. Le même type de comportement, mais plus atténué, peut être observé le 6 septembre autour de 14h. Un des premiers constats techniques, avant la purge et le redémarrage des nœuds de la plate-

forme, était une saturation de la mémoire vive suivie d'une saturation du CPU. L'anomalie la plus frappante se situe dans la nuit du 20 au 21 septembre, où l'on constate que le nombre de message s'est totalement écroulé, et tous les flux de la plateforme CanalTrain sont soudain stoppés. Cette anomalie, qui a dans les faits été résolue très tardivement, provenait du dysfonctionnement du système de gestion des messages (ActiveMQ) interne à CanalTrain. Un redémarrage s'est avéré nécessaire, mais la cause du comportement anormal du composant n'a cependant toujours pas pu être identifiée. On constate qu'avec le paramètre $\varrho \in (0.001, 0.005, 0.01)$ l'anomalie est marquée en début et en fin par deux pics de scores élevés, tandis qu'avec $\varrho = 0.05$ et $\varrho = 0.1$, l'anomalie est signalé du début à la fin. En effet, une valeur de ϱ plus élevée rend le coefficient CFOF plus exigeant en terme de similarité.

Les évaluations présentées dans cette section mettent en avant deux points importants : (1) notre méthode d'approximation du score CFOF à partir de l'arbre iSAX permet d'obtenir une qualité suffisante pour l'application envisagée de détection d'anomalies, (2) le paramètre ϱ permet de régler la sensibilité de la détection, et de l'adapter aux motifs recherchés.

5 Conclusion et perspectives

Nous avons présenté dans cet article une nouvelle méthode permettant d'approcher rapidement le score CFOF d'objets multi-dimensionnels en utilisant un arbre d'indexation iSAX. Nous avons comparé la qualité du score réel avec le score approximé, et montré la très bonne qualité de ce dernier sur un jeu de données artificiel et sur un jeu réel. Par ailleurs, les éléments détectées dans le cas d'utilisation réel au sein de la SNCF reflètent bien un comportement anormal du système d'information validé ultérieurement par les experts.

Plusieurs perspectives sont envisagées pour ce travail. Tout d'abord, bien que l'approximation obtenue soit de bonne qualité, nous souhaitons étudier la possibilité de borner les erreurs commises, et notamment comprendre pourquoi le nombre de dimensions influence le signe de l'erreur lorsque le score augmente. Ensuite, étant donné le faible coût du calcul, nous pensons que notre méthode permet d'une part de traiter les données sous forme de flux, et d'autre part de lever des alertes en temps réel, et de mettre l'arbre à jour incrémentalement. Enfin, il serait intéressant dans notre contexte au sein de la SNCF d'utiliser la temporalité des données pour donner un poids variable aux différents objets lors du calcul du score. Dans notre cas, nous pourrions privilégier les objets similaire du point vue temporel, et par exemple commencer par prendre en compte les objets d'un même type de jour et d'un même créneau horaire, avant d'élargir au besoin aux différents contextes des données.

References

1. M. D. Abràmoff, Y. Lou, A. Erginay, W. Clarida, R. Amelon, . C. Folk, and M. Niemeijer. Improved automated detection of diabetic retinopathy on a publicly

available dataset through integration of deep learning. *Investigative Ophthalmo. and Visual Science*, 57(13):7 pages, 2015.
2. Charu C. Aggarwal. *Outlier Analysis*. Springer Publishing Company, Incorporated, 2013.
3. F. Angiulli. Concentration free outlier detection. In *Machine Learning and Knowledge Discovery in Databases*, pages 3–19. Springer International Publishing, 2017.
4. Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: Identifying density-based local outliers. *SIGMOD Rec.*, 29(2):93–104, May 2000.
5. V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection for discrete sequences: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 24(5):823–839, May 2012.
6. Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58, July 2009.
7. Edwin M Knox and Raymond T Ng. Algorithms for mining distance-based outliers in large datasets. In *Proc. of the International Conference on Very Large Data Bases*, pages 392–403, 1998.
8. X. Li, J. Han, S. Kim, and H. Gonzalez. Roam: Rule- and motif-based anomaly detection in massive moving object data sets. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, pages 273–284, 2007.
9. S. Mukkamala, G. Janoski, and A. Sung. Intrusion detection using neural networks and support vector machines. In *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290)*, volume 2, pages 1702–1707, May 2002.
10. J. Shieh and E. Keogh. iSAX: Indexing and mining terabyte sized time series. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 623–631, New York, NY, USA, 2008. ACM.
11. J. Shieh and E. Keogh. iSAX: disk-aware mining and indexing of massive time series datasets. *Data Mining and Knowledge Discovery*, 19(1):24–57, Aug 2009.
12. K. M. Ting, F. T. Liu, and Z. Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining(ICDM)*, volume 00, pages 413–422, 12 2008.