Etude des méthodes de détection d'anomalies non supervisées appliquées aux flux de données

Kévin Ducharlet, Louise Travé-Massuyès, Marie-Véronique Le Lann, Youssef Miloudi

Etude des méthodes de détection d'anomalies non supervisées appliquées aux flux de données

K. Ducharlet[1,2], L. Travé-Massuyès[2], M-V. Le Lann[2], Y. Miloudi[1]

[1] Carl Berger-Levrault
[2] LAAS-CNRS, Université de Toulouse, CNRS, INSA

{kevin.ducharlet, youssef.miloudi}@carl.eu
{louise, mvlelann}@laas.fr

Résumé

La détection d'anomalies est un sujet d'intérêt en fouille de données. Ces dernières années, le nombre d'applications reposant sur des flux continus de données se multiplie. Ces flux de données sont accompagnés de spécificités dont les algorithmes de détection d'anomalies doivent tenir compte. Pour cette raison, de nombreuses méthodes adaptées à la détection en ligne ont vu le jour. Cet article présente un tour d'horizon des méthodes de détection non supervisées appliquées aux flux de données et discute de l'insuffisance de métriques permettant d'évaluer et comparer ces méthodes.

Mots-clés

Détection d'anomalies, Flux de données, Etat de l'art, Fouille de données

Abstract

Outlier detection is a subject of interest in data mining. During the last decade, the amount of domains featuring data streams grew a lot. Those data streams come with peculiarities that outlier detection methods have to deal with. That is why various methods adapted to the online outlier detection problem have been developed. This article presents a survey on outlier detection unsupervised methods for data streams and discusses the lack of metrics allowing to evaluate and compare these methods.

Keywords

Outlier detection, Data streams, Survey, Data mining

1 Introduction

La détection d'anomalies est un sujet de recherche d'intérêt dans le cadre de la fouille de données qui touche de nombreux domaines d'applications. Les anomalies peuvent être une source d'information importante ou une nuisance à retirer. Dans tous les cas, les détecter est souvent crucial.

Depuis le début des années 2000, les travaux dans le contexte des flux de données se sont multipliés [1, 30]. En effet, les données sont générées sous forme de flux continus dans de nombreux domaines d'application (réseaux de capteurs, surveillance de l'activité web, étude de données météorologiques, surveillance du trafic réseau, ...). Ces données étant souvent sensibles à l'apparition d'anomalies, il est naturel de vouloir y appliquer des méthodes de détection d'anomalies. Cependant, les flux de données dénotent des spécificités nouvelles auxquelles les méthodes doivent s'adapter [37].

Récemment, un grand nombre de méthodes ont vu le jour pour détecter des anomalies dans les flux de données. L'objectif de cet article est de réaliser un tour d'horizon de l'état de l'art de ce domaine. Dans une première partie, nous décrivons brièvement le problème de la détection d'anomalies. Une seconde partie décrit les spécificités des flux de données. Les sections suivantes présentent les différents types de méthodes pouvant s'appliquer à la détection d'anomalies dans les flux de données. Il est important de noter que ces types ne constituent pas des ensembles disjoints ou incompatibles mais permettent de bien saisir les principes des différentes approches existantes. Enfin, nous discutons du manque d'approches comparatives pour ces méthodes avant de conclure cet article.

Travaux connexes. De nombreuses études ont été réalisées afin de recenser et classifier les méthodes de détection d'anomalies [13, 36]. Certaines d'entre elles mentionnent le problème de la détection en ligne en citant quelques méthodes [47, 41, 45] mais sans en faire un tour d'horizon suffisamment complet.

Il existe tout de même des études spécialisées dans la détection d'anomalies dans les flux de données. Certaines d'entre elles traitent d'une catégorie de méthodes ou un cas d'application spécifique. Tran, Han et Shahabi [44], par exemple, se concentrent sur les méthodes qui utilisent la distance entre les points pour déterminer les anomalies. A notre connaissance, les seules études se concentrant sur la détection d'anomalies dans les flux de données et qui en réalisent un tour d'horizon suffisamment complet sont celle de Thakkar, Vala et Prajapati [43] et celle de Salehi et Rashidi [39].

2 Détection d'anomalies

La détection d'anomalies est un sujet de recherche qui a intéressé différentes communautés depuis la fin du 19ème siècle, à commencer par les statisticiens comme en té-

moignent les travaux de Edgeworth [16]. Aussi, différentes définitions ont été fournies pour désigner le terme "anomalie" selon le domaine d'étude mais aussi le domaine d'application, si bien qu'il est impossible d'en donner une définition unique. Néanmoins, la définition qui revient le plus fréquemment dans la littérature est celle de Hawkins [19] : une anomalie est une observation qui s'écarte tant des autres observations qu'on puisse supposer qu'elle ait été générée par un mécanisme différent. On en distingue en général trois types [13] :

— les anomalies ponctuelles, qui correspondent à des points paraissant anormaux par rapport au reste du jeu de données ;
— les anomalies contextuelles, qui sont des points anormaux dans le contexte, temporel et/ou spatial, dans lequel ils apparaissent ;
— les anomalies collectives, qui sont des points normaux individuellement mais anormaux quand considérés comme un ensemble.

Avec plus d'un siècle de travaux dans le domaine, de nombreuses méthodes ont vu le jour. On les sépare principalement selon les informations qu'elles requièrent lors d'une phase d'apprentissage [13] :

— les méthodes supervisées nécessitent, lors de l'apprentissage, un label étiquetant chaque point comme normal ou anormal ;
— les méthodes semi-supervisées n'apprennent que sur des données labellisées comme normales et peuvent ensuite déterminer si un nouveau point est similaire au jeu d'entraînement (normal) ou s'il est différent (anormal) ;
— les méthodes non supervisées n'ont besoin d'aucun label, leur précision est néanmoins inférieure à celle des autres méthodes et il peut être nécessaire de faire des hypothèses fortes, en fixant par exemple le taux d'anomalies attendu.

Pour chaque observation évaluée, la sortie des méthodes de détection d'anomalies peut être de deux types : 1) dans le cas des méthodes supervisées, on obtient souvent une décision (normal ou anormal), 2) dans les autres cas, le degré d'anomalie d'une observation est évalué à partir d'un score ; on peut cependant se ramener à une décision binaire en appliquant un seuil sur ce score.

Nous traiterons ici du cas non supervisé. En effet, les informations nécessaires aux méthodes supervisées ou semi-supervisées sont rarement disponibles, en particulier dans le cas des flux de données, pour lesquels donner un label au fil de l'acquisition est difficile.

3 Spécificités et difficultés des flux de données

Définition Un flux de données est un jeu de données $\mathcal{D} := \{d_t, t \geq 0\}$ de taille infinie où chaque élément d_t correspond à un couple $d_t := (\tau_t, \mathbf{x}_t)$ d'une valeur p-variée $\mathbf{x}_t$ horodatée par une date unique τ_t. Ce flux est généré par une source avec une périodicité pouvant, selon le cadre d'application, ne pas être fixe ; pour $i \neq j$ et $i, j > 0$, on peut avoir $\tau_i - \tau_{i-1} \neq \tau_j - \tau_{j-1}$. Enfin, à chaque instant t, on ne dispose que d'un flux partiel $\mathcal{D}_t := \{d_i, t - \alpha \leq i \leq t - 1, \alpha \geq 1\}$ de points antérieurs pour évaluer d_t.

Par définition, les flux de données sont proches des séries temporelles. Nous considérons ici la subtilité que l'analyse des *séries temporelles* a pour objectif de prédire les observations à venir à partir d'un unique apprentissage sur les données passées, tandis que les *flux de données* introduisent l'idée d'un flux continu avec la nécessité d'un *apprentissage en ligne ou incrémental*.

Spécificités Sept spécificités des flux de données sont formulés dans l'état de l'art [37] :

— *Etat éphémère* : chaque point d_t a une durée de vie déterminée ; l'intérêt du point n'étant pas durable, il doit être traité par la méthode de détection d'anomalies dès qu'il est généré dans le flux de données.
— *Temporalité* : chaque point d_t étant associé à une date τ_t, la notion d'anomalie définie par le modèle doit tenir compte du contexte temporel des points ; dans les flux de données, on ne cherche pas d'anomalies ponctuelles car l'étude est toujours faite dans un contexte défini.
— *Infinité* : les données sont générées en continu, $\mathcal{D}$ est donc de taille infinie et les approches classiques consistant à stocker tous les points avant de générer le modèle et de rechercher les anomalies ne peuvent pas s'appliquer ; dans le cadre des flux de données, les méthodes doivent travailler sur une représentation sommaire du jeu partiel $\mathcal{D}_t$ et cette représentation sommaire doit pouvoir être incrémentée avec les nouveaux points entrants.
— *Vitesse de génération* : les points arrivant en continu et devant être traités dès qu'ils arrivent, il est nécessaire que le temps d'exécution de la classification d'un nouveau point et de l'incrémentation du modèle soit inférieur à la durée entre l'arrivée de deux points consécutifs. De plus, si la vitesse de génération est variable, alors la vitesse d'exécution doit pouvoir s'y adapter quitte à réduire la précision des résultats en travaillant avec une représentation plus réduite pour accélérer le calcul.
— *Non-stationnarité* : la distribution des données peut évoluer à travers le temps ; les méthodes faisant l'hypothèse d'une distribution fixe ne sont donc pas applicables.
— *Incertitude* : dans certains cas d'application, comme les réseaux de capteurs, les mesures générées ne sont pas fiables car elles peuvent être perturbées par des phénomènes environnementaux ; les mesures de similarité utilisées doivent tenir compte de cette incertitude. Celle-ci touche également la vitesse de génération des points qui peuvent être relevés avec du retard ou ne pas l'être du tout ; dans ce cas, il faut tout de même maintenir l'évaluation des points dans leur contexte temporel.
— *Multi-dimensionnalité* : les flux de données sont sujets aux problèmes usuels en grandes dimensions ;

ces problèmes rendent certaines méthodes d'estimation de densité peu efficaces et forcent l'utilisation de mesures de similarité adaptées.

Il existe également d'autres spécificités dans le cas du traitement de plusieurs flux de données [37], comme par exemple dans l'étude du trafic réseau entre plusieurs appareils. Il faut alors considérer les corrélations entre les flux, leur caractère asynchrone (rendant la contextualisation sur plusieurs flux difficile), l'aspect dynamique des relations (dû au comportement asynchrone et à la non-stationnarité des flux individuels) et enfin l'hétérogénéité des flux (les variables mesurées ne sont pas nécessairement les mêmes). Nous ajoutons également que, afin de respecter les spécificités mentionnées plus tôt, le calcul est désormais souvent embarqué, notamment dans le cadre des réseaux de capteurs. Cette spécificité impose des limites sur l'utilisation CPU et de l'espace mémoire des méthodes embarquées.

4 Adaptation par fenêtrage

Les premières approches pour la détection d'anomalies dans les flux de données ont cherché à adapter les méthodes déjà existantes pour des jeux de données statiques dans un cadre dynamique avec une composante temporelle. Pour ce faire, des fenêtres glissantes ont été utilisées afin de ne prendre en compte qu'une partie restreinte et évolutive du jeu de données, permettant ainsi d'adresser une grande partie des spécificités des flux de données.

Il existe plusieurs approches pour le fenêtrage [39] :

- le fenêtrage par point de repère, entre un point de repère fixé dans le jeu de données et la dernière observation générée ;
- le fenêtrage glissant, pour lequel on fixe la taille de la fenêtre (durée ou nombre d'échantillons) puis on la fait glisser à chaque nouvelle observation ;
- le fenêtrage amorti, où on associe à chaque point un poids selon son ancienneté, ainsi les points les plus récents auront un poids plus élevé que les plus anciens ;
- le fenêtrage adaptatif, similaire à un fenêtrage glissant mais où la taille de la fenêtre dépend de la vitesse à laquelle les données évoluent. La fenêtre sera grande si la distribution est stable et petite si la distribution évolue rapidement.

Néanmoins, à chaque fois que la fenêtre est modifiée, le modèle appris doit être mis à jour en conséquence. Les méthodes les plus adaptées sont donc celles qui ne nécessitent pas de réaliser un nouvel apprentissage complet sur la fenêtre actualisée.

5 Méthodes par erreur de prédiction

Nous avons mentionné en introduction de cette étude le lien entre les flux de données et les séries temporelles. Aussi, les méthodes généralement utilisées pour la détection d'anomalies dans des séries temporelles ont été utilisées pour traiter des flux de données, comme présenté dans la récente étude comparative de Duraj et Szczepaniak [15].

L'analyse des séries temporelles se concentre sur l'identification de tendances (changements de comportements linéaires au cours du temps) et de comportements cycliques ou saisonniers afin de définir une relation entre les observations passées et les observations futures. A partir de ces relations, il est donc possible de prédire les prochaines valeurs. Les méthodes de détection d'anomalies s'appuient sur l'erreur de prédiction ; plus l'erreur de prédiction est grande, plus l'observation est éloignée du modèle et peut être considérée comme anormale.

Exemples de méthodes. Parmi les méthodes reposant sur l'erreur de prédiction, nous citerons :

- les modèles ARIMA (Auto-Regressive Integrated Moving Average), dont la méthodologie est détaillée dans le livre de Asteriou et Hall [7] ;
- les modèles de prédictions utilisant le lissage exponentiel ou EST (Exponential Smoothing State Space Model) [22] ;
- les LSTM (Long Short-Term Memory) [29], réseaux de neurones inspirés des réseaux récursifs prenant en entrée les valeurs passées pour prédire les valeurs futures.

Avantages et inconvénients. La limite des modèles de prédiction vient de la caractéristique non-stationnaire des données. En général, la relation est déterminée à partir de données d'entraînement puis appliquée sans ajustement possible. Mettre à jour le modèle est souvent coûteux et difficile à mettre en place en suivant la contrainte de vitesse de génération des flux de données. Notons aussi que les LSTM, comme la majorité des méthodes utilisant des réseaux de neurones, sont peu adaptés à l'apprentissage en ligne à cause de la complexité même du modèle.

6 Approches de partitionnement dynamique

Les approches de partitionnement, ou clustering, sont des techniques, généralement non supervisées, qui ont pour objectif de regrouper les points dans l'espace selon leur similarité [13] et, à ce titre, elles sont liées au méthodes basées distance décrites en Section 8. Ces méthodes ne sont initialement pas pensées pour la détection d'anomalies mais plusieurs d'entre elles ont historiquement été utilisées à cette fin en faisant l'une des trois hypothèses suivantes :

- les points normaux appartiennent à des groupes, ou clusters, contrairement aux anomalies ; pour pouvoir détecter des anomalies, les méthodes de partitionnement ne doivent donc pas forcer tous les points à appartenir à un groupe (exemple : DBSCAN [17]),
- les points normaux sont proches du plus proche centroïde de cluster tandis que les anomalies en sont éloignées ; il faut calculer l'emplacement des centroïdes, barycentres des points de chaque groupe, et la distance des points aux centroïdes. Néanmoins, si un ensemble d'anomalies forment un groupe isolé, alors ces anomalies seront considérées comme normales (exemple : Smith et al. [40]),
- les points normaux appartiennent à des clusters

denses et de grande taille tandis que les anomalies appartiennent à de petits clusters épars ; contrairement à la seconde hypothèse, les anomalies doivent former des clusters isolés et le nombre de groupes à former doit donc être important (exemple : FindCBLOF [20]).

Dans un contexte statique, les méthodes de partitionnement réalisent un seul apprentissage sur les données puis placent les nouveaux points dans les clusters appris en fonction de leur distance aux centroïdes. Cette approche n'est cependant pas viable dans le cas des flux de données à cause de la non-stationnarité de la distribution. Des approches de partitionnement dynamique ont ainsi été développées pour permettre aux clusters d'évoluer dans le temps.

Exemples de méthodes. Il existe de nombreuses méthodes de partitionnement dynamique, présentées en détails dans différentes études [39, 47]. Nous n'en citerons ici que quelques unes :

- BIRCH [48] et CluStream [2] utilisent des caractéristiques des clusters (CFs) contenant, pour chaque cluster, le nombre de points contenus ainsi que la somme et la somme des carrés des valeurs ;
- DenStream [11], DStream [14] et SDstream [35] sont des améliorations de CluStream modifiant respectivement le paramétrage du nombre de clusters, la séparation des clusters en grille et le regroupement en micro-clusters ;
- DyClee [9] travaille avec des micro-clusters, regroupés en clusters selon leur densité et la distance entre eux, et permet de rejeter des micro-clusters anormaux.

Avantages et inconvénients. Les méthodes de partitionnement dynamique citées résument les caractéristiques des clusters avec un nombre fini de métriques. Cette approche permet de limiter le temps nécessaire pour chercher dans quel groupe se positionnent les nouveaux points et, dans la plupart des cas, facilite l'incrémentation du modèle. Pour les CFs par exemple, ajouter un nouveau point à un cluster nécessite simplement d'incrémenter de un le nombre de points et d'ajouter la valeur du point à la somme des valeurs et la valeur du carré à la somme des carrés. Il n'est donc pas nécessaire de stocker l'entièreté du jeu de données.

Cependant, les méthodes de partitionnement dynamique sont souvent critiquées dans le cas de la détection d'anomalies car leur premier objectif est de regrouper les points et non de détecter des anomalies [43]. Aussi, les méthodes les plus adaptées sont celles qui : 1) ne nécessitent pas de fixer le nombre de clusters comme paramètre et 2) sont capables de créer de nouveaux clusters pouvant être considérés anormaux.

7 Méthodes statistiques

L'approche statistique fait l'hypothèse que les données ont été générées par une distribution statistique. L'objectif de ces méthodes est alors d'estimer empiriquement la distribution statistique en question. Les points normaux apparaissent dans des zones de l'espace où la densité de probabilité est élevée tandis que les anomalies apparaissent dans des zones de faible densité de probabilité.

Les méthodes statistiques sont généralement séparées en deux catégories : les méthodes paramétriques et les méthodes non-paramétriques [47].

7.1 Méthodes paramétriques

Les méthodes paramétriques font l'hypothèse que les données suivent une distribution prédéfinie. Les données à disposition sont ensuite utilisées pour déterminer, de manière empirique, les paramètres de ce modèle en minimisant ou maximisant une métrique choisie.

Un exemple typique est celui des modèles Gaussiens, où l'objectif est de déterminer la moyenne et l'écart-type qui maximisent la vraissemblance. Les modèles à base de mélanges gaussiens (GMM) [8] sont aussi populaires pour la détection d'anomalies et ont notamment été utilisés sur des séries temporelles, couplées à un modèle de régression linéaire [3].

Cependant, puisque les méthodes paramétriques font l'hypothèse que les données suivent une distribution fixée, elles ne sont pas applicables dans le cadre des flux de données [43].

7.2 Méthodes non-paramétriques

Il existe principalement deux catégories de méthodes statistiques non-paramétriques. A l'opposé des méthodes paramétriques, il n'est pas nécessaire de faire d'hypothèses a priori concernant la distribution.

7.2.1 Construction d'histogrammes

En statistiques, les histogrammes sont généralement utilisés pour obtenir une représentation visuelle d'une distribution empirique à une dimension. L'espace est divisé en cellules pour lesquelles des colonnes sont construites. La hauteur des colonnes correspond au nombre d'échantillons dont la valeur tombe dans la cellule. Ainsi, une cellule associée à une forte probabilité aura une colonne plus haute qu'une cellule de faible probabilité. La forme de l'histogramme tend vers celle de la fonction de densité de la distribution quand le nombre d'échantillons grandit. On peut donc naturellement utiliser la hauteur de la cellule pour identifier les anomalies.

Exemples de méthodes. Il existe trois types d'approches pour ces méthodes [47] :

- construction à partir des données normales (semi-supervisée) : avec cette approche, les anomalies sont les points qui tombent dans des cellules vides ;
- construction à partir des anomalies (semi-supervisée) : les anomalies tombent cette fois dans des cellules non-vides, cette approche est principalement utilisée dans des cas d'applications où il n'existe qu'un nombre fini de profils anormaux connus ;
- construction sans labels (non supervisée) : on définit les cellules anormales comme celles dont le nombre d'éléments est inférieur à un seuil, dépendant du nombre d'éléments dans les autres cellules de l'histogramme et de la taille de la cellule ; les

échantillons dans ces cellules anormales sont considérés comme anormaux.

Pour le cas multivarié, il est commun de construire un histogramme par variable puis de calculer un score sous forme d'agrégation de la probabilité estimée sur chaque variable. On peut notamment citer HBOS [18] qui calcule son score comme

$$HBOS(x) = \sum_{i=1}^{p} log(\frac{1}{hist_i(x)})$$

où p est le nombre de variables, x l'échantillon à évaluer et $hist_i(x)$ est la hauteur de la cellule dans laquelle tombe x pour la i-ième variable.

Avantages et inconvénients. Parmi les avantages de ces méthodes, nous pouvons noter qu'elles sont faciles à implémenter mais également faciles à incrémenter en recalculant la hauteur des cellules avec de nouveaux points. Cependant, elles deviennent rapidement limitées quand le nombre de dimensions augmente [47].

7.2.2 Méthodes à noyaux

L'approche non-paramétrique la plus connue est celle de l'estimation de densité par noyau (KDE) ou méthode de Parzen-Rosenblatt [32]. Celle-ci est proche de la construction d'histogrammes mais avec une notion de continuité et elle permet d'obtenir une approximation empirique de la fonction de densité de probabilité associée à la distribution. Formellement, soit $x_1, x_2, ..., x_n$ n échantillons i.i.d. (indépendants et identiquement distribués) d'une variable aléatoire X, l'estimateur de la fonction de densité f est

$$\tilde{f}_h(x) = \frac{1}{Nh} \sum_{i=1}^{n} K(\frac{x - x_i}{h})$$

où K est la fonction noyau (on choisit souvent le noyau Gaussien ou le noyau de Epanechnikov) et h est un paramètre jouant sur la zone d'influence de chaque échantillon, ou autrement dit sur le lissage de la courbe. Choisir un h trop faible génère une courbe où chaque échantillon est représenté par un pic de densité tandis que choisir un h trop grand engendre une courbe trop lisse.

On peut citer [26] parmi les méthodes qui ont cherché à adapter cette approche aux spécificités des flux de données.

Avantages et inconvénients. Les méthodes à noyaux ont l'avantage de donner une meilleure approximation de la densité que les méthodes à base d'histogrammes pour un nombre limité d'observations. La notion de continuité corrige aussi une partie des problèmes liés à l'augmentation du nombre de dimensions. Néanmoins, le paramètre h est connu comme étant sensible à paramétrer pour obtenir de bons résultats, et la complexité de la méthode augmente toujours rapidement avec le nombre de variables.

8 Méthodes s'appuyant sur la distance

Ces méthodes utilisent la distance entre les échantillons dans l'espace pour calculer leur score d'anomalie. Plus un point est isolé et plus il est anormal. La grande majorité des méthodes appartenant à cette catégorie généralisent la notion des plus proches voisins (kNN) aux flux de données en travaillant dans des fenêtres glissantes [44]. Il convient également de citer la méthode des HalfSpaceTrees (HST) [42] qui, d'une certaine manière, adapte la méthode des forêts d'isolation [28] aux flux de données.

8.1 Méthodes basées sur les kNN

Présentation des kNN. Les méthodes des plus proches voisins reposent sur la définition d'anomalie donnée par Knorr et Ng [24]. Selon cette définition, un point est anormal selon un critère de distance (DB-anormal) si la proportion de points du jeu de données se trouvant à une distance supérieure à D est au moins r. On parle alors d'un point $DB(r, D)$-anormal. Cette notion est ensuite simplifiée pour considérer comme anormaux tous les points ayant moins de k voisins à une distance inférieure ou égale à d.

De nombreuses approches découlent de cette définition en étudiant certaines statistiques des k plus proches voisins (kNN) comme la somme des distances [5] ou certaines statistiques du k-ième plus proche voisin (k^{th}NN) comme sa distance seule [34].

Exemples d'adaptations en ligne. Pour adapter l'approche des plus proches voisins aux flux de données, dont la taille n'est pas limitée, les méthodes décrites ici utilisent des fenêtres glissantes. Cette approche facilite la recherche des voisins proches.

L'étude comparative de Tran, Fan et Shahabi [44] décrit et compare cinq de ces méthodes, à savoir : exact-Storm et approx-Storm [6], Abstract-C [46], DUE et MCOD [25], ainsi que Thresh_LEAP [12]. Ces méthodes proposent différentes approches pour indexer les données. Ces stuctures indexées facilitent les trois étapes cruciales des kNN en ligne : retrouver les voisins proches, retirer des points de la structure lorsqu'ils sortent de la fenêtre glissante et en ajouter de nouveaux.

La conclusion de cette étude comparative des méthodes de détection en ligne reposant sur la distance entre les points est que MCOD offre de meilleures performances en général. Il est intéressant de noter que les méthodes ne sont pas comparées selon leur précision mais selon leur temps d'exécution et la mémoire utilisée, des critères importants dans le cas des flux de données.

Avantages et inconvénients. Ces méthodes sont pensées pour répondre aux différentes spécificités des flux de données (état éphémère, infinité, vitesse de génération, non-stationnarité). Cependant, les méthodes se basant sur la distance sont sujettes au fléau de la dimension. De plus, les performances des méthodes reposant sur des fenêtres dépendent grandement du choix de la taille de la fenêtre.

8.2 HST

Méthode des forêts d'isolation. L'algorithme des forêts d'isolation [28] adapte les forêts aléatoires à la détection d'anomalies. On construit un arbre d'isolation en choisissant aléatoirement, à chaque embranchement, une variable et une valeur selon laquelle réaliser une séparation. Chaque noeud contient donc un certain nombre d'obser-

vations. La séparation s'arrête lorsque chaque feuille de l'arbre ne contient qu'un unique point. Intuitivement, plus un point a été rapidement isolé (faible hauteur dans l'arbre), plus il est anormal. On construit ainsi un ensemble d'arbres (forêt) aléatoirement et on calcule pour chaque point la hauteur moyenne à laquelle il est isolé. En pratique, il n'est pas nécessaire de construire l'arbre complet.

Adaptation en ligne Les HST [42] sont une forme d'adaptation des forêts d'isolation au problème de la détection en ligne. La différence dans la construction des arbres vient du fait que seule la dimension à séparer est choisie aléatoirement ; la valeur est quant-à-elle prise au milieu de l'intervalle contenant les points de la dimension retenue. On utilise ensuite des fenêtres consécutives et on évalue les points d'une fenêtre par rapport à l'arbre construit dans la fenêtre précédente.

Avantages et inconvénients. Le modèle de mise à jour des HST est particulièrement rapide et s'adapte donc bien aux spécificités de la détection en ligne. Cependant, les résultats dépendent grandement du choix de la taille des fenêtres consécutives. Des fenêtres trop petites ne permettent pas d'avoir une bonne représentation de la répartition des points dans l'espace tandis que des fenêtres trop grandes induisent un temps de retard dans l'adaptation du modèle en cas de changement de distribution.

9 Méthodes s'appuyant sur la densité

Cette section recense les méthodes généralisant le LocalOutlierFactor (LOF) [10] à l'apprentissage en ligne, avec notamment le LOF incrémental (iLOF) [33]. Elles sont étroitement liées aux méthodes statistiques non-paramétriques en ce sens que le LOF tend vers la densité de probabilité quand le nombre d'échantillons augmente.

Présentation du LOF. Le LOF est une mesure d'anomalie reposant sur la densité locale qui utilise les kNN. Cette mesure est construite à partir de la moyenne du rapport de la concentration de points autour des plus proches voisins d'un point par rapport à la concentration de points autour de ce point. Si les plus proches voisins d'un point x sont dans une zone de l'espace très dense par rapport à la densité de la zone de l'espace dans laquelle se trouve x, alors le rapport moyen de concentration sera élevé ; on obtiendra donc un LOF, mesure d'anomalie, élevé.

iLOF : une adaptation incrémentale. Il est possible de prouver qu'ajouter ou supprimer un point d'un jeu de données n'influence qu'une petite partie de ses plus proches voisins dans le calcul du LOF [33]. iLOF se base sur cette propriété pour rendre le LOF incrémental et adapté à la détection en ligne. La précision de la méthode est similaire à celle obtenue en entraînant un nouveau modèle à chaque fois qu'un point est ajouté, mais en limitant considérablement le temps de calcul.

Variantes. Néanmoins, plusieurs méthodes ont été publiées pour améliorer les performances de iLOF : I-IncLOF [23], MiLOF [38], DILOF [31], TADILOF [21] et GP-LOF [4].

Avantages et inconvénients. A l'image des méthodes s'appuyant sur la distance, ces méthodes respectent une grande partie des spécificités des flux de données. Cependant, comme le prouve le nombre de travaux cherchant à améliorer iLOF, il est difficile de réduire le temps de calcul de ces méthodes.

10 Discussion sur l'évaluation des méthodes en ligne

En parcourant l'état de l'art de la détection d'anomalies en ligne, nous avons noté qu'il n'existait que très peu d'études comparatives de ces méthodes.

L'étude comparative de Duraj et Szczepaniak [15] ne compare qu'un nombre très limité de méthodes tout en mentionnant à quel point la complexité des flux de données et la diversité des cas d'application rendaient toute comparaison difficile. De même, celle de Tran, Fan et Shahabi [44] ne compare que les méthodes de distance et uniquement selon des critères de performance algorithmique.

La première difficulté est en réalité de pouvoir évaluer les méthodes de détection en ligne sur des critères mettant en avant les spécificités des flux de données. A notre connaissance, la contribution la plus complète sur l'évaluation des méthodes en ligne est le Numenta Anomaly Benchmark (NAB) [27]. Le NAB propose une métrique pour évaluer la capacité des méthodes à détecter les anomalies dans une série temporelle en récompensant la détection en amont de l'anomalie labellisée et en pénalisant les faux positifs et faux négatifs. Cette approche nécessite néanmoins des jeux de données labellisés représentatifs du domaine d'application dans lequel les méthodes seront appliquées.

11 Conclusion

Au-delà de la grande diversité de méthodes cherchant à répondre à la problématique de la détection d'anomalies dans les flux de données, cet état de l'art a principalement permis d'identifier deux points :

- proposer une méthode répondant aux spécificités des flux de données n'est pas une tâche aisée ; les contraintes traitées sont l'*état éphémère* des points et la *non-stationnarité*, qui sont les plus cruciales, mais aussi l'*infinité* et la *vitesse de génération* ;
- le domaine manque d'une méthode d'évaluation qui permettrait de vérifier à quel point chaque méthode répond à *chacune des spécificités* et de les comparer entre elles.

Références

[1] D. Abadi, D. Carney, U. Çetintemel, M. Cherniack, C. Convey, C. Erwin, E. Galvez, M. Hatoun, A. Maskey, A. Rasin, A. Singer, M. Stonebraker, N. Tatbul, Y. Xing, R. Yan, and S. Zdonik. Aurora : A data stream management system. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, SIGMOD '03, page 666, New

York, NY, USA, 2003. Association for Computing Machinery.

[2] Charu C. Aggarwal, Philip S. Yu, Jiawei Han, and Jianyong Wang. - a framework for clustering evolving data streams. In Johann-Christoph Freytag, Peter Lockemann, Serge Abiteboul, Michael Carey, Patricia Selinger, and Andreas Heuer, editors, *Proceedings 2003 VLDB Conference*, pages 81–92. Morgan Kaufmann, San Francisco, 2003.

[3] Hermine N. Akouemo and Richard J. Povinelli. Probabilistic anomaly detection in natural gas time series data. *International Journal of Forecasting*, 32(3) :948–956, 2016.

[4] Raed Alsini, Omar Alghushairy, Xiaogang Ma, and Terrance Soule. A grid partition-based local outlier factor for data stream processing. In Hamid R. Arabnia, Ken Ferens, David de la Fuente, Elena B. Kozerenko, José Angel Olivas Varela, and Fernando G. Tinetti, editors, *Advances in Artificial Intelligence and Applied Cognitive Computing*, pages 1047–1060, Cham, 2021. Springer International Publishing.

[5] F. Angiulli and C. Pizzuti. Outlier mining in large high-dimensional data sets. *IEEE Transactions on Knowledge and Data Engineering*, 17(2) :203–215, Feb 2005.

[6] Fabrizio Angiulli and Fabio Fassetti. Detecting distance-based outliers in streams of data. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, page 811–820, New York, NY, USA, 2007. Association for Computing Machinery.

[7] Dimitros Asteriou and Stephen G Hall. Arima models and the box–jenkins methodology. *Applied Econometrics*, 2(2) :265–286, 2011.

[8] David Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

[9] Nathalie Barbosa Roa, Louise Travé-Massuyès, and Victor Hugo Grisales. DyClee : Dynamic clustering for tracking evolving environments. *Pattern Recognition*, 94 :162–186, October 2019.

[10] 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.

[11] Feng Cao, Martin Estert, Weining Qian, and Aoying Zhou. Density-based clustering over an evolving data stream with noise. In *Proceedings of the 2006 SIAM international conference on data mining*, pages 328–339. SIAM, 2006.

[12] Lei Cao, Di Yang, Qingyang Wang, Yanwei Yu, Jiayuan Wang, and Elke A. Rundensteiner. Scalable distance-based outlier detection over high-volume data streams. In *2014 IEEE 30th International Conference on Data Engineering*, pages 76–87, March 2014.

[13] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection : A survey. *ACM Comput. Surv.*, 41(3), jul 2009.

[14] Yixin Chen and Li Tu. Density-based clustering for real-time stream data. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, page 133–142, New York, NY, USA, 2007. Association for Computing Machinery.

[15] Agnieszka Duraj and Piotr S. Szczepaniak. Outlier Detection in Data Streams — A Comparative Study of Selected Methods. *Procedia Computer Science*, 192 :2769–2778, 2021.

[16] F.Y. Edgeworth. Xli. on discordant observations. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 23(143) :364–375, 1887.

[17] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press, 1996.

[18] Markus Goldstein and Andreas Dengel. Histogram-based outlier score (hbos) : A fast unsupervised anomaly detection algorithm. *KI-2012 : poster and demo track*, 9, 2012.

[19] Douglas M Hawkins. *Identification of outliers*, volume 11. Springer, 1980.

[20] Zengyou He, Xiaofei Xu, and Shengchun Deng. Discovering cluster-based local outliers. *Pattern Recogn. Lett.*, 24(9–10) :1641–1650, jun 2003.

[21] Jen-Wei Huang, Meng-Xun Zhong, and Bijay Prasad Jaysawal. Tadilof : Time aware density-based incremental local outlier detection in data streams. *Sensors*, 20(20), 2020.

[22] Rob J Hyndman, Anne B Koehler, Ralph D Snyder, and Simone Grose. A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting*, 18(3) :439–454, 2002.

[23] Seyed Hesamodin Karimian, Manouchehr Kelarestaghi, and Sattar Hashemi. I-inclof : Improved incremental local outlier detection for data streams. In *The 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP 2012)*, pages 023–028, May 2012.

[24] Edwin M. Knorr and Raymond T. Ng. Algorithms for mining distance-based outliers in large datasets. In *Proceedings of the 24rd International Conference on Very Large Data Bases*, VLDB '98, page 392–403, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[25] Maria Kontaki, Anastasios Gounaris, Apostolos N. Papadopoulos, Kostas Tsichlas, and Yannis Manolopoulos. Continuous monitoring of distance-based out-

liers over data streams. In *2011 IEEE 27th International Conference on Data Engineering*, pages 135–146, April 2011.

[26] Matej Kristan, Aleš Leonardis, and Danijel Skočaj. Multivariate online kernel density estimation with gaussian kernels. *Pattern Recognition*, 44(10) :2630–2642, 2011. Semi-Supervised Learning for Visual Content Analysis and Understanding.

[27] Alexander Lavin and Subutai Ahmad. Evaluating real-time anomaly detection algorithms – the numenta anomaly benchmark. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pages 38–44, Dec 2015.

[28] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422, Dec 2008.

[29] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, Puneet Agarwal, et al. Long short term memory networks for anomaly detection in time series. In *Proceedings*, volume 89, pages 89–94, 2015.

[30] Rajeev Motwani, Jennifer Widom, Arvind Arasu, Brian Babcock, Shivnath Babu, Mayur Datar, Gurmeet Singh Manku, Chris Olston, Justin Rosenstein, and Rohit Varma. Query processing, approximation, and resource management in a data stream management system. In *First Biennial Conference on Innovative Data Systems Research, CIDR 2003, Asilomar, CA, USA, January 5-8, 2003, Online Proceedings*. www.cidrdb.org, 2003.

[31] Gyoung S. Na, Donghyun Kim, and Hwanjo Yu. Dilof : Effective and memory efficient local outlier detection in data streams. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '18, page 1993–2002, New York, NY, USA, 2018. Association for Computing Machinery.

[32] Emanuel Parzen. On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics*, 33(3) :1065 – 1076, 1962.

[33] Dragoljub Pokrajac, Aleksandar Lazarevic, and Longin Jan Latecki. Incremental local outlier detection for data streams. In *2007 IEEE Symposium on Computational Intelligence and Data Mining*, pages 504–515, March 2007.

[34] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. *SIGMOD Rec.*, 29(2) :427–438, may 2000.

[35] Jiadong Ren and Ruiqing Ma. Density-based data streams clustering over sliding windows. In *2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, volume 5, pages 248–252, Aug 2009.

[36] Lukas Ruff, Jacob R. Kauffmann, Robert A. Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G. Dietterich, and Klaus-Robert Müller. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 109(5) :756–795, May 2021.

[37] Shiblee Sadik and Le Gruenwald. Research issues in outlier detection for data streams. *SIGKDD Explor. Newsl.*, 15(1) :33–40, mar 2014.

[38] Mahsa Salehi, Christopher Leckie, James C. Bezdek, Tharshan Vaithianathan, and Xuyun Zhang. Fast memory efficient local outlier detection in data streams. *IEEE Transactions on Knowledge and Data Engineering*, 28(12) :3246–3260, Dec 2016.

[39] Mahsa Salehi and Lida Rashidi. A survey on anomaly detection in evolving data : [with application to forest fire risk prediction]. *SIGKDD Explor. Newsl.*, 20(1) :13–23, may 2018.

[40] R. Smith, A. Bivens, M. Embrechts, C. Palagiri, and Boleslaw Szymanski. Clustering approaches for anomaly based intrusion detection. *Proceedings of Intelligent Engineering Systems Through Artificial Neural Networks*, pages 579–584, 01 2002.

[41] SS Sreevidya et al. A survey on outlier detection methods. *IJCSIT) International Journal of Computer Science and Information Technologies*, 5(6), 2014.

[42] Swee Chuan Tan, Kai Ming Ting, and Tony Fei Liu. Fast anomaly detection for streaming data. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, IJCAI'11, page 1511–1516. AAAI Press, 2011.

[43] Pooja Thakkar, Jay Vala, and Vishal Prajapati. Survey on outlier detection in data stream. *International Journal of Computer Applications*, 136 :13–16, 02 2016.

[44] Luan Tran, Liyue Fan, and Cyrus Shahabi. Distance-based outlier detection in data streams. *Proc. VLDB Endow.*, 9(12) :1089–1100, aug 2016.

[45] Hongzhi Wang, Mohamed Jaward Bah, and Mohamed Hammad. Progress in outlier detection techniques : A survey. *IEEE Access*, 7 :107964–108000, 2019.

[46] Di Yang, Elke A. Rundensteiner, and Matthew O. Ward. Neighbor-based pattern detection for windows over streaming data. In *Proceedings of the 12th International Conference on Extending Database Technology : Advances in Database Technology*, EDBT '09, page 529–540, New York, NY, USA, 2009. Association for Computing Machinery.

[47] Ji Zhang. Advancements of outlier detection : A survey. *EAI Endorsed Transactions on Scalable Information Systems*, 1(1), 2 2013.

[48] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch : An efficient data clustering method for very large databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, SIGMOD '96, page 103–114, New York, NY, USA, 1996. Association for Computing Machinery.