Découverte de sous-groupes de prédictions interprétables pour le triage d'incidents

Youcef Remil*,***, Anes Bendimerad***, Marc Plantevit**
Céline Robardet*, Mehdi Kaytoue*,***

*Université de Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205, F-69621, France
**EPITA Research and Development Laboratory (LRDE), France
***Infologic, 99 avenue de Lyon, F-26500, Bourg-Lès-Valence, France
yre@infologic.fr

Résumé. Le besoin de maintenance prédictive s'accompagne d'un nombre croissant d'incidents qui doivent être rapidement assignés aux services appropriés pour des actions correctives. Il existe des modèles prédictifs pour automatiser cette assignation, mais les plus efficaces sont opaques. Des méthodes ont été conçues pour expliquer localement chaque prédiction de tels modèles, mais elles fournissent une explication à chaque résultat, inconcevable en présence d'un nombre important de prédictions à analyser. Nous proposons d'abord un modèle efficace de triage d'incidents, puis une méthode basée sur la découverte de sous-groupes pour grouper les explications de ses prédictions. Cette méthode permet (1) de grouper les incidents dont les prédictions partagent des explications similaires et (2) de fournir une description interprétable à chacun de ces sous-groupes d'incidents. Cet article est une traduction résumée de (Remil et al., 2021).

1 Introduction

De nos jours, de nouvelles technologies sont constamment introduites et de nouvelles perspectives de numérisation sont adoptées en industrie : des logiciels comme l'Enterprise Resource Planning (ERP) sont au centre de cette révolution. Il s'ensuit que la maintenance de ces systèmes logiciels est devenue de plus en plus complexe car impliquant toujours plus de composants physiques, logiciels et métiers interdépendants. En effet, les incidents sont de diverses natures : (i) un incident prédit à partir des modèles construits sur des données historiques, (ii) un incident réel non encore signalé par l'utilisateur final mais par le système de supervision et (iii) finalement, un incident rapporté par l'utilisateur final, mais généralement mal décrit et contextualisé. Lorsque le nombre d'incidents remontés est important, il devient difficile pour les ingénieurs d'astreinte de les évaluer rapidement et les orienter vers les services appropriés pour mener des actions correctives. Afin d'accélérer et d'améliorer ces décisions, plusieurs modèles prédictifs ont été proposés pour le *triage d'incidents*, dont le but est de prédire efficacement pour un incident signalé le service à cibler. Les meilleures approches de l'état de l'art utilisent des modèles d'apprentissage profond (Chen et al., 2019; Pham et al., 2020; Wang et al., 2021). Toutefois, ces modèles sont opaques et difficiles à interpréter puisqu'ils ne révèlent pas la logique derrière leur mécanisme de prise de décision. Un défi important pour

réussir à automatiser le triage d'incidents est d'abord d'assurer la confiance des ingénieurs en ces modèles, en leur fournissant des explications sur leurs prédictions. De nombreuses recherches ont été menées pour interpréter ces modèles (dits, *boîtes noires*) afin de profiter de leur haute performance tout en offrant un processus décisionnel transparent à l'utilisateur. Ces méthodes peuvent être caractérisées comme globales versus locales. Les méthodes globales visent à expliquer la logique interne du modèle, en apprenant un modèle potentiellement interprétable (e.g., arbres de décision, règles de classification) qui imite les prédictions du modèle original (Hara et Hayashi, 2018; Han et al., 2015). Cependant, ces méthodes sont généralement moins fidèles au modèle s'il est extrêmement complexe. Un autre axe de recherche vise à expliquer indépendamment chaque prédiction du modèle et en fournir une explication locale. Plusieurs méthodes ont été proposées dans cette direction (Ribeiro et al., 2016; Guidotti et al., 2019a; Simonyan et al., 2014). Cependant, lorsque le nombre de résultats à expliquer est grand, il devient difficile d'analyser individuellement ces explications. Peu de méthodes ont été proposées pour résoudre ce problème, que ce soit en regroupant des explications similaires en clusters (Ibrahim et al., 2019) ou en sélectionnant un sous-ensemble d'explications représentatives (Ribeiro et al., 2016). Si elles peuvent ainsi capturer les différentes explications, ces deux méthodes ont cependant en commun la limitation de ne pas fournir à l'utilisateur le contexte dans lequel chaque explication se tient. Dans cet article, nous présentons d'abord un modèle de triage d'incidents efficace basé sur 170k incidents signalés sur plus de mille serveurs d'applications au cours des 7 dernières années sur notre ERP Infologic Copilote. Ensuite, nous proposons une nouvelle approche basée sur la *découverte de sous-groupes* (Atzmueller, 2015), qui permet de regrouper les objets prédits en sous-groupes qui supportent la même explication. Chaque sous-groupe est également associé à une description qui le caractérise de manière unique. Cette approche adresse d'une nouvelle façon le problème d'explication des résultats du modèle, en particulier lorsque le nombre de résultats à expliquer est important.

2 Méthodologie

Nous conduisons notre analyse sur un ensemble R de 170k incidents remontés sur plus de 1000 serveurs où chaque serveur contient une instance du logiciel Copilote. Nous pré-traitons le texte décrivant l'incident pour extraire les termes discriminants. Nous étendons également les incidents avec divers attributs permettant de contextualiser les incidents. Un *rapport d'incident* de $R = \{r_1, ...r_n\}$ est décrit par : (1) le titre r_{titre}, (2) la description r_{desc} qui contient du texte décrivant l'incident, (3) le composant $r_{\text{composant}}$ dans lequel l'incident est arrivé, (4) le moment de création de l'incident r_{temps}. Les rapports d'incidents sont augmentés avec d'autres informations comme les variables d'environnement du système, des indicateurs de performance et alertes générées à un temps proche de celui de l'incident. Pour chaque incident, nous effectuons la *tokenisation*, le *stemming* et la *lemmatisation* de r_{titre} et r_{desc}. Ensuite, nous calculons les coefficients `tf-idf` et conservons les 10K termes les plus discriminants.

Modèle de données. Nous considérons le jeu de données défini par : (O, A, Y), où $O = \{o_i\}_{1 \leqslant i \leqslant n}$ est un ensemble d'objets qui font référence aux incidents, $A = (a_j)_{1 \leqslant j \leqslant m}$ est un vecteur d'attributs descriptifs, et les classes $Y = \{y_1, ..., y_n\}$ représentent les services assignés à chaque incident. Chaque attribut a se modélise par une fonction $a : O \longrightarrow R_a$ avec R_a le domaine de l'attribut a pouvant être *numérique*, *nominal*, ou *booléen*. Un incident est assigné

O	$\mathbf{r_{titre}}$		$\mathbf{r_{desc}}$			$\mathbf{r_{temps}}$	**Variables ENV**		**Alertes**	**Métriques**	**Service**
	a_1 disk	a_2 swap	a_3 full	a_4 java	a_5 http	a_6 weekend	a_7 Soft. ver.	a_8 Soft. type	a_9 Memory usage	a_{10} % used heap	y classe
o_1	0.7	0	0.4	0	0	Vrai	1	Ventes	-	60	**TEC**
o_2	0	0.8	0.3	0	0	Vrai	3	Ventes	Bloquant	50	**TEC**
o_3	0.5	0	0	0	0.6	Vrai	2	Superviseur	-	60	**TEC**
o_4	0	0.5	0.9	0.6	0	Vrai	3	Superviseur	Critique	97	**OT**
o_5	0	0.7	0.6	0	0	Faux	1	Ventes	Critique	96	**OT**
o_6	0.1	0	0	0.6	0.6	Faux	2	Ventes	Alarme	85	**OT**
o_7	0.1	0	0	0	0.9	Faux	1	Ventes	-	60	**OT**

TAB. 1: *Exemple de référence d'un jeu de données* (O, A, Y).

à un service (une classe) $y \in \{Class_1, ..., Class_p\}$ avec p le nombre total de services. Ces notations sont illustrées dans la Table 1 avec un jeu de données contenant 7 incidents $O = \{o_1, ..., o_7\}$, chacun est décrit par 10 attributs et une classe y. Les attributs $\{a_1, ..., a_5\}$ sont numériques et représentent le `tf-idf` des termes qui caractérisent r_{titre} et r_{desc}.

Triage d'incidents. Le triage d'incidents consiste à prédire la classe y (le service) pour un incident r en fonction de ses attributs $\{a_1, ..., a_m\}$. Nous considérons des sous-ensembles d'apprentissage et de validation $O_T, O_V \subseteq O$ utilisés pour entraîner différents modèles notés par b. La prédiction $b(o) = \{b(o)_1, ..., b(o)_p\} \in [0, 1]^p$ fournit une distribution de probabilité sur les p classes prédites. Inspirés par des travaux récents (Pham et al., 2020; Gao et al., 2020) qui montrent que les meilleurs résultats sont généralement obtenus par les modèles sophistiqués, nous avons évalué les méthodes suivantes : Random Forest, XGBoost et plusieurs architectures de réseaux de neurones profonds. Nous avons toutefois considéré la régression logistique et le multinomial Naive Bayes comme des modèles de référence interprétables. Notre étude comparative a montré que les meilleures performances ont été atteintes par un modèle de réseaux de neurones profond. Nous référons à (Remil et al., 2021) pour plus de détails sur cette étude. Nous adoptons donc ce modèle pour le triage d'incidents.

Explication locale des résultats. Ce problème consiste à donner une explication e pour la prédiction $b(o) = \{b(o)_1, ..., b(o)_p\} \in [0, 1]^p$ qui concerne un incident spécifique $o \in O$ où e appartient à un domaine interprétable E (Guidotti et al., 2019b).

Modèle boîte blanche. Afin d'extraire une explication $e \in E$ pour une prédiction $b(o)$, une des méthodes les plus populaires consiste à entraîner un modèle interprétable qui apprend à imiter les prédictions de b spécifiquement dans le voisinage de l'objet o. Ce modèle interprétable est appelé *boîte blanche* et noté par w. Nous utilisons la régression Ridge comme modèle interprétable. Ce choix est justifié par sa capacité de régularisation dans notre cas qui consiste en un grand nombre d'attributs dépassant le nombre d'objets à expliquer, ainsi que la forte corrélation qui existe entre ces attributs.

Voisinage local. Pour un objet $o \in O$ donné, on note $N(o)$ l'ensemble des objets générés synthétiquement au voisinage de o, plus l'objet o. Pour expliquer o, nous entraînons un modèle w qui imite b sur l'ensemble $N(o)$.

Exemple. Nous montrons dans la Table 2 la probabilité de la classe TEC qui indique le service technique (resp., OT qui indique le service outils), prédite par le modèle b. Pour chaque objet $o \in O$, un modèle local w a été entraîné pour imiter le comportement de b dans le voisinage

O	**Pred.** $b(o)_1$ **de TEC**	**Pred.** $b(o)_2$ **de OT**	**Modèle local** $w(o)$ **de la classe majoritaire**	**Modèle de sous-groupe**	**Description du sous-groupe**
o_1	**0.9**	0.1	$\text{disk} + 0.1 \cdot \text{swap} + 0.5 \cdot \text{full}$	$0.8 \cdot \text{disk} + 0.7 \cdot \text{swap} + 0.4 \cdot \text{full}$	weekend = True $\wedge$ java = 0
o_2	**0.8**	0.2	$0.1 \cdot \text{disk} + \text{swap} + 0.4 \cdot \text{full}$		
o_3	**0.6**	0.4	$\text{disk} - 0.5 \cdot \text{http} + 0.3 \cdot \text{full}$		
o_4	0.3	**0.7**	$0.9 \cdot \text{java} + 0.5 \cdot \text{full} - 0.2 \cdot \text{swap}$	$\text{java} + 0.6 \cdot \text{full} - 0.3 \cdot \text{swap}$	%used heap $\geqslant 96$
o_5	0.2	**0.8**	$\text{java} + 0.6 \cdot \text{full} - 0.3 \cdot \text{swap}$		
o_6	0.2	**0.8**	$0.8 \cdot \text{http}$	$0.9 \cdot \text{http}$	Soft. type = Sales $\wedge$ weekend = False
o_7	0.1	**0.9**	$0.9 \cdot \text{http} - 0.1 \cdot \text{disk}$		

TAB. 2: *Regroupement des explications associées aux prédictions des incidents du Tableau 1*

de o. Par exemple, le modèle local qui fournit une explication de la prédiction de TEC pour o_3 est : $w(o) = \text{disk} - 0.5 \cdot \text{http} + 0.3 \cdot \text{full}$. Cela indique que les termes *disk* et *full* contribuent à la probabilité d'assigner l'incident au service TEC, contrairement au terme *http*. En pratique, on peut avoir un grand ensemble $O_E \subseteq O$ d'incidents dont les prédictions $\{b(o) \mid o \in O_E\}$ doivent être expliquées. Fournir une explication spécifique pour chaque prédiction est fastidieux pour l'utilisateur qui doit étudier chaque explication séparément. De plus, de nombreux objets qui ont certaines propriétés en commun peuvent avoir des explications similaires. Nous cherchons donc à réduire le nombre d'explications en partitionnant les objets en sous-groupes $s \subseteq O_E$ pour lesquels on peut fournir une explication commune en les caractérisant avec une description. Nous optons pour les concepts de *la découverte de sous-groupes*.

3 Découverte de sous-groupes de prédictions interprétables

La découverte de sous-groupes (Atzmueller, 2015) cherche à identifier des sous-groupes interprétables qui favorisent certaines propriétés d'intérêt par rapport à un problème cible. Bien que cette technique ait été largement utilisée dans divers domaines (e.g., physique, neurosciences), elle n'a pas été encore exploitée dans le cadre de XAI. Une telle adaptation nécessite de relever plusieurs défis liés à la structure de données, la mesure de qualité du sous-groupe ainsi qu'à l'algorithme d'exploration qui optimise cette nouvelle mesure.

Langage de motifs. Un *motif* d est un *sélecteur* d'un sous-ensemble d'objets à partir de l'ensemble de données en utilisant leurs valeurs d'attributs. Nous nous référons à l'ensemble de tous les motifs possibles par le *langage des motifs* $\mathcal{D}$. Dans notre cas, $\mathcal{D} = \times_{i=1}^{m} \mathcal{D}_i$ où $\mathcal{D}_i$ est donné par l'ensemble de tous les intervalles possibles dans $\mathbb{R}$ si a_i est numérique, l'ensemble $\{C_i, \varnothing\} \cup \{\{c\} \mid c \in C_i\}$ si a_i est nominal, ou $\{\{0,1\},\{0\},\{1\}\}$ si a_i est booléen. Un motif $d \in \mathcal{D}$ est par conséquent donné par un ensemble de restrictions sur chaque attribut (i.e. $d = (d_i)_{1 \leqslant i \leqslant m}$). Ces motifs sont ordonnés du plus général au plus restrictif par une relation d'ordre $\sqsubseteq$. Pour deux motifs $c = (c_i)_{1 \leqslant i \leqslant m} \in \mathcal{D}$ et $d = (d_i)_{1 \leqslant i \leqslant m} \in \mathcal{D}$, on a : $c \sqsubseteq d \Leftrightarrow \forall i \in [\![1, m]\!] \, (c_i \supseteq d_i)$.

Motifs et objets. On parle d'un motif $d = (d_i)_{1 \leqslant i \leqslant m}$ qui *couvre* un objet $o \in O_E$ ssi $\forall i \in [\![1, m]\!] : a_i(o) \in d_i$. Un motif d *couvre* un ensemble $O' \subseteq O_E$ ssi il couvre chaque objet $o \in O'$. En utilisant le concept de couverture, nous définissons une fonction $\delta(O') \in \mathcal{D}$ qui donne le motif le plus restrictif qui couvre un ensemble d'objets O' : $\forall d \in \mathcal{D}$, d *covers* O' ssi $d \sqsubseteq \delta(O')$. Pour un motif d donné, l'ensemble de tous les objets couverts par d est référencé par $ext(d) = \{o \in O_E \mid d \sqsubseteq \delta(o)\}$. Dans la Table 1, par simplification, nous considérons un

sous-ensemble de 4 attributs $A' =$ (java, weekend, Soft. type, %used heap). Un exemple de motif dans A' est $d' = (\mathbb{R}^+, Vrai, \text{Ventes}, [0, 100])$. Ce motif d' ne couvre que $O' = \{o_1, o_2\}$. Le motif le plus restrictif pour O' dans A' est $\delta(O') = (0, Vrai, \text{Ventes}, [50, 60])$, et nous avons $d' \sqsubseteq \delta(O')$.

Sous-groupe. Un sous-groupe est un sous-ensemble d'objets $s \subseteq O_E$ qui peut être sélectionné en utilisant les motifs d sur les attributs A, et on note $\mathcal{S} = ext(\mathcal{D}) = \{ext(d) \mid d \in \mathcal{D}\}$.

Au lieu de fournir une explication pour chaque prédiction d'un objet, nous cherchons à regrouper ces objets en un petit nombre K de sous-groupes qui couvrent tous les objets de O_E. Pour chaque sous-groupe, nous fournissons une explication commune à tous les objets du sous-groupe. Comme illustré dans la Table 2, la prédiction de chaque objet o est expliquée par un modèle local interprétable w entraîné au voisinage de o. On peut partitionner les données en trois sous-groupes dont les objets peuvent avoir la même explication. Par exemple, le premier sous-groupe fait référence à tous les incidents qui se sont produits le week-end et qui ne contiennent pas le mot java dans leur description. Leur probabilité prédite de TEC peut s'expliquer par la même relation : $0.8 \cdot \text{disk} + 0.7 \cdot \text{swap} + 0.4 \cdot \text{full}$. Pour garantir qu'un modèle de sous-groupe est valable pour tous les objets du sous-groupe, nous cherchons à minimiser l'erreur entre les prédictions de b et les résultats du modèle de sous-groupe w_s. En d'autres termes, nous cherchons à maximiser la fidélité du modèle de sous-groupes par rapport au modèle d'origine.

Modèle de sous-groupes. C'est un modèle de boîte blanche utilisé pour expliquer les prédictions de b sur les objets d'un sous-groupe s. Il est entraîné sur les voisinages de tous les objets de s.

Fonction d'erreur. Nous utilisons la somme des erreurs au carré pour évaluer la fidélité d'un modèle de sous-groupe interprétable w_s, ajusté sur un sous-groupe s et son voisinage d'objets :

$$L(s, w_s, b) = \sum_{o \in s} \sum_{o' \in N(o)} \sum_{i=1}^{p} \left(b(o')_i - w_s(o')_i\right)^2 .$$

L'erreur globale pour l'ensemble des sous-groupes $S = \{s_1, s_2, ...\} \subseteq \mathcal{S}$ avec leurs modèles $W = \{w_{s_1}, w_{s_2}, ...\}$ est défini comme : $L(S, b) = \sum_{i=1}^{|S|} L(s_i, w_{s_i}, b)$.

Afin de contrôler le nombre total des explications collectives des prédictions $\{b(o) \mid o \in O_E\}$, nous proposons de définir un seuil maximal $K \in \mathbb{N}$ pour le nombre de sous-groupes retournés. L'objectif est donc de trouver un ensemble de sous-groupes $\{s_1, s_2, ...\}$ de cardinalité au plus K, dont les modèles associés $\{w_{s_1}, w_{s_2}, ...\}$ minimisent l'erreur globale.

Problème 1 (Regroupement des explications) *Soit $O_E \subseteq O$ un sous-ensemble d'objets dont on cherche à expliquer les prédictions, et b le modèle boîte noire utilisé. Étant donné un seuil $K \in \mathbb{N}$ spécifié par l'utilisateur qui représente le nombre maximum d'explications, trouver un ensemble de sous-groupes $S = \{s_1, s_2, ...\}$ avec leurs modèles associés $W = \{w_{s_1}, w_{s_2}, ...\}$ tel que (1) $|S| \leqslant K$, (2) l'ensemble des sous-groupes couvre tous les objets de O_E : $\bigcup_{s \in S} s = O_E$, et (3) l'erreur globale est minimisée : $S = argmin_{S' \subseteq \mathcal{S}} L(S', b)$.*

4 Algorithme `SplitSD4X`

Le problème de regroupement des explications avec la découverte de sous-groupes est NP-difficile. Nous avons donc opté pour une stratégie heuristique mais qui a empiriquement montré son efficacité dans l'étude que nous avons menée. Cet algorithme commence d'abord par générer un voisinage local $N(o)$ pour chaque objet $o \in O_E$ en utilisant un processus d'échantillonnage que nous détaillons dans (Remil et al., 2021). Ensuite, il construit au fur et à mesure un ensemble de sous-groupes qui ne se chevauchent pas, en utilisant une stratégie basée sur la séparation. L'algorithme commence par l'ensemble de sous-groupes $S = \{O_E\}$ qui contient un seul sous-groupe couvrant tous les objets de O_E. À chaque itération, et étant donné l'ensemble actuel de sous-groupes S, l'un des sous-groupes de S est divisé en deux sous-groupes qui minimisent l'erreur globale. La séparation est appliquée pour un seul attribut $a \in A$ à chaque fois qui vérifie la contrainte d'optimalité de l'erreur globale. Cette procédure est effectuée de manière itérative jusqu'à ce que le nombre de sous-groupes K soit atteint, ou jusqu'à ce qu'il n'y ait aucune amélioration possible de l'erreur globale. Le détail complet de l'algorithme est fourni dans (Remil et al., 2021).

5 Expérimentations

Cette étude expérimentale cherche à évaluer l'efficacité de `SplitSD4X` à regrouper les explications des prédictions d'un modèle boîte noire entraîné spécifiquement pour le triage d'incidents. Nous visons à répondre aux questions suivantes : (Q1) les modèles de sous-groupes fournissent-ils de bonnes explications, i.e., les explications des sous-groupes sont-elles *fidèles* aux prédictions de la boîte noire ? (Q2) les modèles de sous-groupes sont-ils *humainement interprétables* et aident-ils les praticiens à comprendre les résultats de la boîte noire ?

Nous avons collecté 170k incidents impliquant plus de 1k serveurs au cours des 7 dernières années chez Infologic. Nous considérons un ensemble d'apprentissage (65%), validation (10%) et ensemble de test (25%). La répartition des incidents entre les différents 19 services qu'on reconnaît est extrêmement déséquilibrée, les services les plus populaires ont des milliers d'incidents tandis que d'autres services ont rarement été sollicités. Nous sélectionnons aléatoirement dans l'ensemble de tests 2000 incidents pour les regrouper au maximum en 200 sous-groupes avec leurs explications. Ainsi, nous appliquons `SplitSD4X` avec une taille de voisinage de 250 pour chaque objet ($|N(o)| = 250$). Le processus complet nécessite environ 3 heures pour s'exécuter lorsque le nombre de sous-groupes K est fixé à 200. Tout au long de ces expériences, nous comparons `SplitSD4X` avec un **modèle global interprétable** (`global-wb`), et un **modèle local interprétable** (`local-wb`). Le premier consiste à entraîner un modèle interprétable sur l'ensemble des données à expliquer ainsi que leurs voisinages pour approximer globalement les prédictions de la boîte noire. Pour la seconde, nous entraînons un modèle interprétable local pour expliquer chaque objet indépendamment. Nous suivons la même méthodologie appliquée pour LIME (Ribeiro et al., 2016), néanmoins, nous optons pour notre méthode de génération de voisinage local proposée pour pouvoir comparer entre `SplitSD4X` et `local-wb`.

Q1 : Pour une première analyse, nous utilisons l'erreur quadratique moyenne $\texttt{MSE} = \frac{\texttt{SSE}}{|O_E|}$ entre les prédictions faites par `SplitSD4X` et la boîte noire b par rapport au nombre de sous-groupes calculés K. Les résultats sont donnés dans la Fig. 1a où nous rapportons éga-

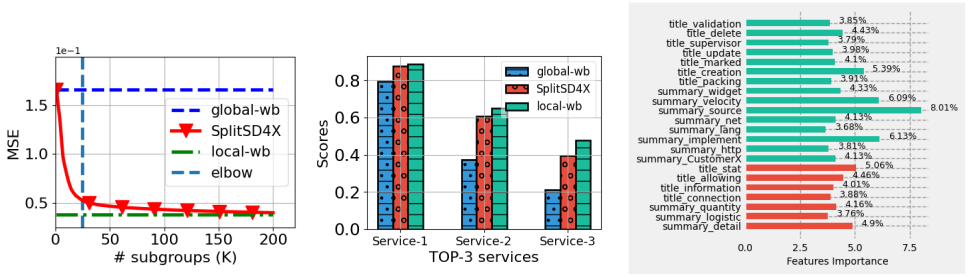

FIG. 1: *Qualité des explications des résultats de la boîte noire : (a)* `MSE` *avec différentes valeurs de* K*, (b)* `F1-score` *avec* $K^\star$ *trouvé par* `elbow`*, (c) l'importance des attributs pour la prédiction du service "Ventes".*

lement deux valeurs constantes : le `MSE` obtenu par les méthodes `global-wb` et `local-wb`. Lorsque le nombre de sous-groupes augmente, le `MSE` devient brusquement plus petit. Le gain le plus important est obtenu avec seulement un petit nombre de sous-groupes. Pour trouver le nombre optimal de sous-groupes $K^\star$, de sorte que la fidélité soit très proche de celle obtenue par le `local-wb`, et qu'augmenter davantage K n'améliore pas significativement la fidélité, nous utilisons la technique elbow. Nous montrons qu'avec seulement 25 modèles de sous-groupes, nous pouvons améliorer la fidélité de `global-wb` et obtenir un score assez proche de celui de `local-wb` qui utilise 2000 modèles locaux. Dans une seconde analyse, nous évaluons le `F1-score` sur les 3 services les plus probables obtenus par chaque méthode par rapport aux résultats de la boîte noire. Par exemple, pour **service-2**, nous prenons les services ayant une seconde meilleure probabilité et nous les comparons aux seconds services probables prédits par la boîte noire. Notre solution est évaluée sur uniquement 25 modèles de sous-groupes. Les résultats sont présentés dans la Figure 1b. Les scores obtenus par `SplitSD4X` sont toujours significativement meilleurs que ceux de `global-wb`. De plus, avec seulement 25 sous-groupes, nous obtenons des résultats presque similaires à ceux de `local-wb` (0,87 pour `SplitSD4X` et 0,88 pour `local-wb` pour **service-1**). Ces résultats démontrent que `SplitSD4X` est capable de réduire considérablement le nombre d'explications tout en gardant la fidélité par rapport aux prédictions de la boîte noire.

Q2 : À partir des modèles de sous-groupes calculés, nous tirons des explications interprétables qui aident les praticiens à comprendre la raison derrière la prédiction d'un service par rapport à un autre, pour un ensemble d'incidents en identifiant les attributs les plus importants du modèle, i.e., les attributs qui contribuent le plus à la prédiction du service. La Figure 1c montre l'explication correspondante aux prédictions associées à un des sous-groupes identifiés. Ce sous-groupe couvre les incidents déclarés entre 23h et minuit, et qui ne contiennent pas les termes *stats* et *stock* dans leurs descriptions. Nous nous intéressons au service «Ventes» qui a été prédit majoritairement pour ce sous-groupe. La Figure 1c vise donc à expliquer la raison pour laquelle les incidents du sous-groupes sont assignés à «Ventes» par le modèle opaque. L'importance des attributs met en évidence les termes fortement et positivement corrélés au contexte de vente tels que *création*, *mise à jour* et *validation* des commandes. Alors que des termes comme *vélocité* augmentent la probabilité que le service de vente soit sollicité, d'autres termes comme *logistique* et *connexion* diminuent cette probabilité au profit d'autres services.

Références

Atzmueller, M. (2015). Subgroup discovery. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 5*(1), 35–49.

Chen, J., X. He, Q. Lin, H. Zhang, D. Hao, F. Gao, Z. Xu, Y. Dang, et D. Zhang (2019). Continuous incident triage for large-scale online service systems. In *ASE 2019*, pp. 364–375. IEEE.

Gao, J., N. Yaseen, R. MacDavid, F. V. Frujeri, V. Liu, R. Bianchini, R. Aditya, X. Wang, H. Lee, D. A. Maltz, M. Yu, et B. Arzani (2020). Scouts : Improving the diagnosis process through domain-customized incident routing. In *SIGCOMM 2020*, pp. 253–269. ACM.

Guidotti, R., A. Monreale, et L. Cariaggi (2019a). Investigating neighborhood generation methods for explanations of obscure image classifiers. In *PAKDD*, pp. 55–68.

Guidotti, R., A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, et D. Pedreschi (2019b). A survey of methods for explaining black box models. *ACM Comput. Surv. 51*(5).

Han, L., S. Luo, J. Yu, L. Pan, et S. Chen (2015). Rule extraction from support vector machines using ensemble learning approach. *IEEE J-BHI 19*(2), 728–734.

Hara, S. et K. Hayashi (2018). Making tree ensembles interpretable. In *AISTATS*, pp. 77–85.

Ibrahim, M., M. Louie, C. Modarres, et J. W. Paisley (2019). Global explanations of neural networks : Mapping the landscape of predictions. In *AAAI/ACM AIES*, pp. 279–287.

Pham, P., V. Jain, L. Dauterman, J. Ormont, et N. Jain (2020). Deeptriage : Automated transfer assistance for incidents in cloud services. In *ACM SIGKDD*, pp. 3281–3289.

Remil, Y., A. Bendimerad, M. Plantevit, C. Robardet, et M. Kaytoue (2021). Interpretable summaries of black box incident triaging with subgroup discovery. In *IEEE DSAA*.

Ribeiro, M. T., S. Singh, et C. Guestrin (2016). "why should I trust you ?" : Explaining the predictions of any classifier. In *ACM SIGKDD*, pp. 1135–1144.

Simonyan, K., A. Vedaldi, et A. Zisserman (2014). Deep inside convolutional networks : Visualising image classification models and saliency maps. In *ICLR*.

Wang, Y., G. Li, Z. Wang, Y. Kang, Y. Zhou, H. Zhang, F. Gao, J. Sun, L. Yang, P. Lee, et al. (2021). Fast outage analysis of large-scale production clouds with service correlation mining. In *ICSE*, pp. 885–896. IEEE.

Summary

The need for predictive maintenance comes with an increasing number of incidents, where it is imperative to quickly decide which service to contact for corrective actions. Several predictive models have been designed to automate this process, but the efficient models are opaque (say, black boxes). Many approaches have been proposed to locally explain each prediction of such models. However, providing an explanation for every result is not conceivable when it comes to a large number of daily predictions to analyze. In this article we propose a method based on Subgroup Discovery in order to (1) group together objects that share similar explanations and (2) provide a description that characterizes each subgroup.