SANDMAN : un système auto-adaptatif pour la détection d'anomalies dans le flux des données des bâtiments intelligents

Maxime Houssin, Stéphanie Combettes, Marie-Pierre Gleizes, Bérangère Lartigue

SANDMAN : un système auto-adaptatif pour la détection d'anomalies dans le flux des données des bâtiments intelligents

Maxime Houssin[1,2], Stephanie Combettes[1], Marie-Pierre Gleizes[1], Berangere Lartigue[2]
Université Toulouse 3 Paul Sabatier : IRIT[1], LMDC[2]
prenom.nom@{irit.fr[1], univ-tlse3.fr[2]}

Résumé

Actuellement, la gestion de l'énergie au sein des bâtiments est essentielle pour participer à la transition écologique. Pour cela, les bâtiments sont de plus en plus équipés de capteurs pour aider le gestionnaire du bâtiment. Mais l'hétérogénéité et la grande quantité de données générées rend sa tâche assez ardue. Le système multi-agent SANDMAN, décrit dans cet article, a pour objectif d'aider à la détection automatique en temps réel, de plusieurs types d'anomalies en utilisant des données brutes et hétérogènes. SANDMAN fait un apprentissage semi-supervisé en considérant quelques avis d'un expert du domaine. Les résultats montrent que SANDMAN, après une phase d'apprentissage, détecte les différents types d'anomalies, est résistant au bruit et passe l'échelle.

Mots Clef

Système multi-agent auto-adaptatif, Détection d'anomalies, Bâtiments intelligents

Abstract

Currently, energy management within buildings is essential to participate in the green transition. To this end, buildings are increasingly equipped with sensors to assist the building manager. But the heterogeneity and the large amount of data generated makes this task quite difficult. The SANDMAN multi-agent system, described in this paper, aims to assist in the automatic detection, in real time, of several types of anomalies using raw and heterogeneous data. SANDMAN features a semi-supervised learning by considering some feedbacks from an expert in the field. The results show that SANDMAN, after a learning phase, detects the different types of anomalies, is resistant to noise and is scalable.

Keywords

Self-Adaptive Multi-Agent System, Anomaly Detection, Smart Buildings

1 Introduction

Les bâtiments représentent plus de 20% de la consommation mondiale d'énergie, ce chiffre pouvant dépasser 40 % dans les pays développés. Cependant, une grande partie de cette énergie est gaspillée [1] (30 % aux États-Unis et en Europe). Ce gaspillage provient d'une mauvaise gestion des bâtiments comme la non-détection de problèmes.

L'objectif du travail présenté dans cet article concerne la détection d'anomalies dans les données énergétiques au sein de bâtiments intelligents pour mettre en place rapidement des mécanismes pour lutter contre ce gaspillage d'énergie.

Avec l'avènement de *l'IoT (Internet of Things)*, le nombre de capteurs dans les bâtiments existants et nouveaux a fortement augmenté en raison de leur coût moins élevé et de l'avantage évident de leur utilisation pour la gestion des bâtiments. Des capteurs peuvent être facilement ajoutés dans les bâtiments ou remplacés par d'autres. Par conséquent, la gestion de ces capteurs et des données qu'ils génèrent font que les gestionnaires des bâtiments se trouvent face à un système complexe à gérer.

Un contrôle précis de ces données est pourtant nécessaire pour gérer correctement les bâtiments, et notamment leur performance énergétique. C'est pourquoi un outil d'aide à la détection automatique des anomalies est un atout important pour les gestionnaires de bâtiments. Cet outil doit traiter les données **en temps réel** pour pouvoir agir au plus tôt, ceci étant un enjeu essentiel dans la détection des anomalies. Parce que le nombre de données est important, les espaces de recherche des anomalies deviennent aussi très grands, c'est pourquoi le système doit apprendre à lever les anomalies de manière semi-supervisée grâce aux feedbacks d'un expert.

Afin d'établir un état de l'art adapté à notre problématique, nous listons au préalable les caractéristiques qui nous semblent indispensables pour un système de détection d'anomalies, ainsi que les définitions des différents types d'anomalies.

1.1 Caractéristiques requises pour un système de détection d'anomalies

Un bâtiment intelligent équipé de capteurs engendre une grande quantité de données disponibles qui doivent être analysées pour améliorer la gestion d'énergie. Par exemple, le SGE (Service de Gestion et d'Exploitation de l'énergie pour le campus de l'Université P. Sabatier - Toulouse III) gère environ 6000 capteurs mesurant au moins une valeur par heure. Ainsi, la conception de systèmes intelligents de détection d'anomalies dans les bâtiments doit prendre en

compte les caractéristiques suivantes :

- **Interaction avec un expert.** Afin d'informer l'expert sur les anomalies trouvées par le système mais aussi pour avoir des feedbacks de l'expert pour permettre au système d'apprendre tout le long de sa vie.
- **Détection en temps réel.** Afin de réduire l'impact énergétique des anomalies et éventuellement l'inconfort pour les occupants, il est préférable de détecter au plus tôt les anomalies.
- **Détection de plusieurs types d'anomalies.** Pour être le plus exhaustif possible, le système de détection d'anomalies doit être capable de détecter différents types d'anomalies comme les anomalies dues à un seul capteur, des anomalies dues à plusieurs capteurs, des anomalies dues à l'évolution de valeurs de capteurs sur une période donnée et donc qui ne se manifestent qu'au bout d'un certain temps, etc.
- **Utilisation de données brutes.** Compte tenu de la grande quantité de données à manipuler et de la nécessité d'un traitement en temps réel, il convient d'éviter autant que possible le prétraitement des données. En effet, c'est une tâche chronophage, qui peut amener des biais. De plus, le traitement en temps réel laisse peu de temps pour formater les données au fur et à mesure de leur arrivée.
- **Utilisation de données hétérogènes.** Les capteurs au sein d'un bâtiment intelligent mesurent différentes grandeurs physiques (température, pression, consommation d'énergie, etc.) qui doivent toutes être prises en compte pour une meilleure détection des anomalies.
- **Ouverture.** Dans un bâtiment intelligent, des capteurs peuvent être ajoutés et d'autres enlevés. Le système de détection d'anomalies doit continuer à fonctionner dans ces environnements dynamiques.
- **Passage à l'échelle.** Le système de détection d'anomalies doit pouvoir s'appliquer non seulement aux bâtiments, mais également à des ensembles de bâtiments, et à terme, à des campus, des quartiers. Le nombre de capteurs à gérer peut donc être considérable.
- **Généricité.** Pour être utilisable de manière optimale, le système de détection d'anomalies doit pouvoir être déployé dans plusieurs bâtiments sans modification majeure. De plus, dans le domaine des bâtiments intelligents, il existe des systèmes de détection d'anomalies pour le chauffage, d'autres pour la consommation d'électricité, etc. Le développement d'un système unique pour détecter les anomalies dans ces différents domaines est un avantage.

Le système SANDMAN utilise une technique d'apprentissage couplée à l'utilisation de systèmes multi-agents pour prendre en compte ces caractéristiques. Par ailleurs, ce système lève des anomalies à destination d'un *expert* qui doit réagir au mieux. Le système et l'expert sont en interaction : l'expert doit pouvoir voir les levées d'anomalies mais il doit également pouvoir confirmer/infirmer ces anomalies ou en signaler des nouvelles si nécessaire à SANDMAN. Le système SANDMAN est donc un système d'apprentissage semi-supervisé puisqu'il peut recevoir des feedbacks d'un expert. Précisons toutefois que l'expert peut (in)valider une levée d'anomalie *a posteriori* ; il n'est pas tenu d'être constamment actif devant le système et l'apprentissage du système prend en compte cette spécificité.

		sensor1		sensor2		sensor3		sensor4		sensor5		anomaly
row	timestamp	measured	nominal	measured	nominal	measured	nominal	measured	nominal	measured	nominal	
0	2016-01-11T00:00	16	17	11	10	160	160	11	11	66	66	
1	2016-01-11T01:00	15	15	11	11	140	140	-5	11	64	65	A
2	2016-01-11T02:00	15	15	12	12	130	127	10	10	65	65	
3	2016-01-11T03:00	15	16	12	13	110	110	8	8	64	65	
4	2016-01-11T04:00	15	14	12	12	120	120	9	9	65	65	
5	2016-01-11T05:00	15	15	14	14	140	140	13	13	400	65	
6	2016-01-11T06:00	16	17	16	14	170	165	17	17	66	66	
7	2016-01-11T07:00	17	19	20	20	220	200	25	25	67	67	
8	2016-01-11T08:00	23	19	19	23	220	235	34	29	68	67	A
9	2016-01-11T09:00	18	18	25	25	260	255	33	33	68	68	
10	2016-01-11T10:00	19	18	27	27	230	245	35	34	69	69	
11	2016-01-11T11:00	19	19	25	25	210	235	32	32	69	70	
12	2016-01-11T12:00	21	21	26	24	210	230	32	32	71	71	
13	2016-01-11T13:00	22	22	28	28	220	240	34	34	72	72	A
14	2016-01-11T14:00	22	22	29	29	270	270	34	33	72	72	
15	2016-01-11T15:00	23	23	31	31	270	270	35	35	73	73	
16	2016-01-11T16:00	24	24	32	29	270	270	35	35	76	74	
17	2016-01-11T17:00	23	23	29	29	250	258	31	31	73	73	
18	2016-01-11T18:00	22	24	29	29	240	240	31	30	72	72	
19	2016-01-11T19:00	21	22	26	26	220	220	27	27	71	71	
20	2016-01-11T20:00	19	19	22	22	200	200	23	23	69	69	
21	2016-01-11T21:00	19	20	21	22	190	185	21	19	70	69	
22	2016-01-11T22:00	18	18	20	20	180	180	20	20	68	68	
23	2016-01-11T23:00	17	17	16	16	170	170	16	16	67	67	

TABLE 1 – Exemple d'une base de données

1.2 Définition des différents types d'anomalies

Chaque capteur fournit une donnée estampillée avec l'heure de sa perception ou *timestamp*.
Pour chacun, on dispose d'une **valeur réelle** qui correspond à la mesure effectuée et une **valeur nominale** qui est la valeur habituellement fournie par le capteur. Cette valeur nominale est calculée à partir d'un profil. La localisation des capteurs est inconnue, ainsi que toute autre métadonnée (type du capteur, fréquence d'acquisition, etc.). Comme le manque d'information sur les capteurs est un problème courant [2], il est nécessaire de les prendre tous en compte dans la méthode de détection des anomalies.
Comme Chandola et al. [3], nous considérons qu'une anomalie est une situation inattendue ou indésirable dans un système. Ainsi trois types d'anomalies peuvent être pris en compte. Une anomalie est **ponctuelle** si une valeur réelle est en dehors de la plage de valeur acceptable pour le capteur ; une anomalie est **contextuelle** si une valeur réelle se situe dans une plage acceptable pour le capteur mais est anormale dans certains contextes (exemple : consommation de chauffage pendant l'été) ; une anomalie est **collective** si un ensemble de valeurs réelles est anormal par rapport à l'ensemble des données, bien que les valeurs réelles individuelles ne soient pas anormales.
Le tableau 1 illustre les 3 types d'anomalies définies cidessus. Elles sont indiquées par la lettre A dans la dernière colonne. La ligne 2 est considérée comme anormale par l'expert
parce que le capteur 4 a une valeur de -5, ce qui est éloi-

gné de la valeur nominale de 11. Il s'agit d'une anomalie ponctuelle.
La ligne 8 est une anomalie à cause des valeurs des capteurs 1 à 4. Bien que prise individuellement, aucune de ces valeurs ne serait considérée comme une anomalie, toutes ensembles elles génèrent une anomalie collective.
La dernière anomalie de la ligne 13 apparaît à 13 heures et seulement à ce moment-là. Elle est due au fait que les valeurs réelles du capteur 3 sont disparates de la ligne 9 à la ligne 13. Dans la vie réelle, un expert peut avoir remarqué un problème à 13 heures en raison de l'accumulation constante de petites différences dans les 4 lignes précédentes. C'est ce qu'on appelle une anomalie contextuelle car il est nécessaire de prendre en compte le passé récent pour la détecter.
La ligne 5 n'est pas une anomalie même si la valeur de 400 du capteur 5 est très éloignée de la valeur nominale qui est de 65. Elle n'est pas considérée comme une anomalie par l'expert, ce qui signifie que ce capteur est soit sans importance pour la détection des anomalies, soit que la variation n'est pas assez grande pour qu'un expert la considère comme anormale.
L'article se compose de la façon suivante. La section 2 décrit l'état de l'art des principales méthodes existantes de détection des anomalies. La section 3 explicite l'architecture du système multi-agent proposé, SANDMAN (*semi-Supervised ANomaly Detection with Multi-AgeNt systems*). La section 4 est consacrée aux expérimentations et aux résultats de SANDMAN. Enfin, la section 5 conclut et propose des perspectives au travail présenté.

2 Etat de l'art sur la détection d'anomalies dans les bâtiments intelligents

Compte tenu de l'enjeu réel de la détection des anomalies dans les bâtiments, le champ de recherche dans ce domaine est en pleine expansion. Les systèmes multi-agents ont été introduits en physique du bâtiment depuis une dizaine d'années, mais leurs applications est quasiment entièrement dédiées à l'optimisation de la gestion des systèmes (chauffage, refroidissement, ventilation) ou du bâtiment complet [4]. Aucune étude, à notre connaissance, n'a été menée pour une application dans la détection des anomalies dans les données des bâtiments.
Parmi les méthodes utilisées dans la littérature, on trouve les **modèles physiques**, qui nécessitent de modéliser le système ou bâtiment et de le comparer avec les données mesurées. Ces méthodes sont vite limitées, compte tenu de la complexité des systèmes à modéliser [5].
Des méthodes de **classification non supervisée et statistiques** ont été utilisées pour la détection des anomalies dans des données de bâtiment [6] [7] [8]. Cependant, ces études ne portent que sur un faible nombre de données et ne montrent pas que le passage à l'échelle soit possible. Elles reposent également sur l'hypothèse que le nombre d'anomalies est faible par rapport au nombre total de données.

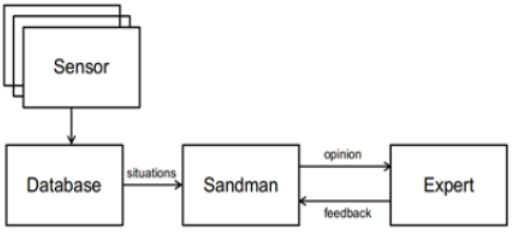

FIGURE 1 – SANDMAN : schéma général

Les **réseaux de neurones** sont des outils largement utilisés dans les études sur les bâtiments, mais rarement pour la détection des anomalies [9], [10]. Même si les réseaux de neurones peuvent détecter des anomalies en temps réel, un grand ensemble de données étiquetées est nécessaire pour entraîner les réseaux. Or, cette étape semble impossible à mener pour les tailles de systèmes considérés ici et elle ne correspond pas à ce qui se passe réellement dans la gestion des bâtiments.
Les méthodes de ***data mining*** sont utilisées pour la détection d'anomalies dans des données de bâtiments [11], [12], [13]. Cependant, une étape de pré-traitement des données est nécessaire, ainsi qu'un expert pour la sélection des données pertinentes lors du passage à l'échelle.
Enfin, des méthodes de **systèmes décentralisés avec multi-agents** ont été utilisées pour la détection d'anomalies dans des données estampillées avec l'heure de leur perception [14][15]. Ces méthodes n'étaient pas appliquées aux bâtiments intelligents, mais pourraient y être adaptées.

3 Architecture de SANDMAN

Cette section présente les principes de base de SANDMAN. La première sous-section décrit les notions de situation et de profil des capteurs, la deuxième explique le fonctionnement du système SANDMAN pour détecter des anomalies et enfin la dernière explicite l'apprentissage. La figure 1 fait référence au fonctionnement général de SANDMAN.

3.1 Définitions : situation et profil d'un capteur

SANDMAN travaille à partir d'informations remontées par un ensemble de capteurs donnés. Par exemple, dans le tableau 1, SANDMAN a accès aux valeurs de 5 capteurs. Nous avons défini la notion de **situation** comme l'ensemble des valeurs mesurées de tous ces capteurs sur 24 heures et on nomme **situation courante** la dernière situation rencontrée. La dernière mesure de tous les capteurs est appelée **valeurs instantanées des capteurs** : ceci correspond dans le tableau 1 à la dernière ligne, contenant les valeurs instantanées des 5 capteurs à 23H00. Les situations rencontrées par SANDMAN sont mémorisées dans un historique des situations.

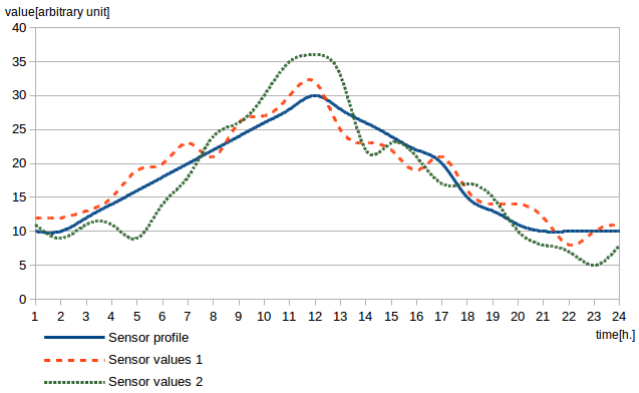

FIGURE 2 – Valeurs mesurées du capteur et profil du capteur en fonction du temps

A chaque capteur est associé un **profil**, constitué de 24 valeurs mesurées, une pour chaque heure de la journée. Ce profil indique la valeur attendue à tout moment de la journée pour le capteur considéré. Il est mis à jour par SANDMAN tout au long de la vie du système.
La figure 2 représente le profil de ce capteur en fonction du temps sur une période de 24 heures et on a représenté par les valeurs mesurées *values 1* un exemple possible de valeurs mesurées pour ce capteur et les valeurs mesurées *values 2* un autre exemple possible de valeurs mesurées pour ce capteur. Les valeurs mesurées *values 1* et *values 2* sont ici deux exemples distincts, une seule valeur pouvant être mesurée à la fois sur un capteur. On peut constater que les valeurs mesurées *values 1* sont proches du profil attendu sur 24 heures alors que les valeurs mesurées *values 2* sont considérées comme anormales.

3.2 Détection d'anomalies par SANDMAN

Etant donné que tous les capteurs disponibles sont utilisés, que l'expert peut ne s'intéresser qu'à certaines anomalies, et que chaque capteur a un intervalle et une amplitude de valeurs qui lui est propre, l'apprentissage des profils ne suffit pas pour détecter les anomalies. En effet, certains capteurs peuvent ne pas être pertinents pour la détection des anomalies. C'est le cas, par exemple, de capteurs situés dans un bâtiment qui n'intéresse pas l'expert, ou bien si l'expert ne s'intéresse qu'à la consommation électrique alors un capteur de qualité de l'air ne sera pas pertinent. Enfin, pour les capteurs qui sont effectivement utiles, selon les capteurs, certains écarts par rapport aux valeurs attendues sont plus importants que d'autres. Ceci est dû au fait que leurs valeurs ne sont pas normalisées car SANDMAN utilise des données brutes.
Aussi, pour lever ou non une anomalie à chaque nouvelle situation courante, SANDMAN utilise à la fois les valeurs mesurées des capteurs sur 24h et les profils des capteurs associés. Pour cela, SANDMAN calcule **la disparité d'un capteur** qui est la somme de toutes les différences entre le profil du capteur et les valeurs mesurées du capteur au cours des dernières 24 heures, c'est-à-dire pour la situation courante (Eq. (1)).

$$Disparity_s^t = \sum_{t_i=t}^{t-23} \left| realValue_s^{t_i} - nominalValue_s^{t_i} \right| \quad (1)$$

avec :

- c : capteur c ;
- t : instant t de la situation courante ;
- $realValue_c^l$: valeur réelle du capteur c au temps l ;
- $nominalValue_c^l$: valeur nominale du capteur c au temps l ;

La période considérée est une fenêtre glissante de 24 heures. Si l'heure de la situation courante est 15h, toutes les données de 16h la veille à 15h le jour courant sont considérées pour le profil allant de 16h à 15h. SANDMAN calcule ensuite **le degré d'anomalie (DA) de la situation courante** qui correspond à la somme des valeurs mesurées instantanées de tous les capteurs, pondérées par un poids (Eq. (2)).

$$DA(Situation^t) = \sum_{c=1}^{S} Disparity_c^t * Poids_c \quad (2)$$

avec :

- $Situation^t$: situation au temps t ;
- S : nombre de capteurs ;
- $Disparity_c^t$: disparité du capteur c calculée au temps t ;
- $Poids_c$: poids associé au capteur c.

Ainsi grâce au degré d'anomalie, SANDMAN peut classer la situation courante, c'est-à-dire de l'étiqueter "normale" ou "anormale" en suivant l'algorithme 1.

Algorithme 1 Algorithme de classification ou de détection d'anomalies

```
pour chaque nouvelle situation courante faire
    si le degré d'anomalie de la situation courante est inférieur à un seuil alors
        SANDMAN retourne : "la situation est normale"
    sinon
        SANDMAN retourne : "la situation est anormale"
    fin si
fin pour
```

Pour détecter les anomalies de manière correcte, il faut donc que les poids $Poids_i$ associés aux capteurs aient les bonnes valeurs. Un capteur a un seul poids associé qui est utilisé pour calculer le degré d'anomalie pour toutes les situations de l'historique des situations.

3.3 Apprentissage par SANDMAN

Une fois que SANDMAN a effectué la classification d'une situation c'est-à-dire qu'il l'a étiquetée "normale" ou "anormale" et qu'il a un retour de l'expert, il effectue un cycle de résolution au cours duquel il analyse s'il doit ou non apprendre. SANDMAN analyse donc les cas suivants :

1. Pour une situation étiquetée "anormale" pour lesquelles SANDMAN et l'expert sont d'accord,
   (a) SANDMAN ne fait rien.
2. Pour une situation étiquetée "anormale" pour l'expert et "normale" pour SANDMAN,
   (a) SANDMAN ajoute à l'historique des situations cette situation étiquetée "anormale"
   (b) SANDMAN lance l'auto-adaptation des poids
3. Pour chaque ligne de la situation étiquetée "normale" pour l'expert et "anormale" pour SANDMAN,
   (a) SANDMAN met à jour les profils des capteurs
   (b) SANDMAN ajoute à l'historique des situations cette situation étiquetée "normale"
   (c) SANDMAN lance l'auto-adaptation des poids
4. Pour chaque ligne de la situation étiquetée "normale" pour l'expert et pour SANDMAN,
   (a) SANDMAN met à jour les profils des capteurs

Les principales étapes concernant la mise à jour des profils, l'ajout d'une situation dans l'historique et l'auto-adaptation des poids sont décrits ci-après.

Mise à jour des profils. Ils ne sont mis à jour que si la situation est étiquetée "normale" par l'expert (étapes 3a et 4a), car les valeurs des capteurs dans les situations anormales ne sont pas fiables et ne représentent pas les valeurs attendues. La formule de mise à jour appliquée à chaque profil de capteur est la suivante (Eq. 3) :

$$val'^{t}_{profil} = (1 - \lambda) * val^{t}_{profil} + \lambda * val^{t}_{capteur} \quad (3)$$

avec

- t : l'heure de la situation courante,
- $\lambda \in]0, 1[$ représente l'importance de la nouvelle valeur par rapport à l'ancienne,
- val^{t}_{profil} : la valeur précédemment apprise et stockée dans le profil pour l'heure t
- $val^{t}_{capteur}$: la valeur mesurée du capteur pour l'heure t
- val'^{t}_{profil} : la nouvelle valeur apprise et stockée dans le profil pour l'heure t

Chaque profil de capteur modifie l'une de ses 24 valeurs correspondant à l'heure t de la situation courante en utilisant la valeur mesurée $val^{t}_{capteur}$ de la situation courante.

Ajout d'une situation à l'historique des situations. Cet ajout, réalisé aux étapes 2a et 3b, lance des cycles de résolution de SANDMAN qui se produisent jusqu'à ce que

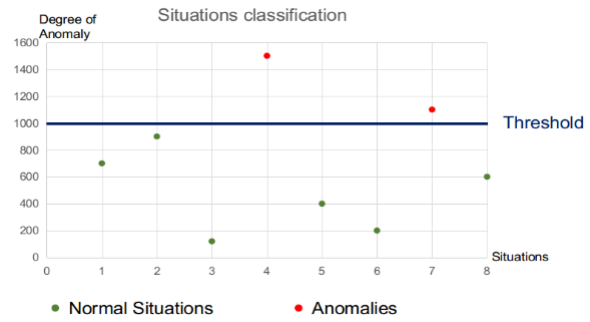

FIGURE 3 – Example of 8 classified situations

toutes les situations présentes dans l'historique soient classées correctement et de manière équilibrée. Une situation est classée correctement si son degré d'anomalie permet de déduire le bon classement (normal ou anormale) de la situation. Les situations sont classées de manière équilibrée si le degré d'anomalie de la pire situation étiquetée "normale" et le degré d'anomalie de la pire situation étiquetée "anormale" sont en valeur absolue à une même distance du seuil donné. Ceci est illustré sur la figure 3, où les points respectivement au dessus et au-dessous du seuil $Threshold$ et les plus près du seuil sont à égale distance de ce seuil.

A chaque situation est attribuée une valeur de criticité en fonction de son degré d'anomalie et de sa classification donnée par l'expert. La criticité représente le degré de satisfaction de la situation par rapport à son objectif. Une situation représentant une situation étiquetée "normale" vise un degré d'anomalie aussi bas que possible par rapport au seuil $Threshold$, tandis qu'une situation étiquetée "anormale" vise un degré d'anomalie élevé par rapport au seuil $Threshold$. La criticité **Crit** est calculée comme suit :

$$Crit_{Situation^t} = DA(Situation^t) - Threshold \quad (4)$$

pour une situation "normale".

$$Crit_{Situation^t} = Threshold - DA(Situation^t) \quad (5)$$

pour une situation "anomalie".

Chaque situation veut minimiser sa criticité, une criticité négative signifie que la situation est correctement classée. Comme les disparités et les poids sont toujours positifs, les situations étiquetées "normales" et les situations étiquetées "anormales" ont des objectifs antagonistes, voulant respectivement des pondérations plus ou moins élevées. Lorsqu'un cycle de résolution est effectué, chaque situation envoie un message à chaque agent poids avec les informations suivantes :

- le sens de changement de poids (plus bas pour une situation étiquetée "normale", plus haut pour une situation étiquetée "anormale") ;

— la criticité de la situation
— l'influence du poids dans le degré d'anomalie.

L'influence d'un poids associé aux capteurs dans une situation correspond à la part du capteur dans le calcul du degré d'anomalie. Elle est calculée comme suit :

$$Influ_w = \frac{Poids_c * Disparity_c^t}{DA(Situation^t)} \quad (6)$$

Un cycle de résolution débute quand les situations demandent aux poids de s'ajuster, le cycle se termine quand les poids se sont auto-adaptés.

Auto-adaptation des poids. Chaque poids est représenté par un agent poids. Le système multi-agent des poids a comme objectif que chaque agent poids trouve sa valeur en coopérant avec les autres. Au cours d'un cycle de résolution, toutes les situations demandent des ajustements de poids (étapes 2b et 3c) et les agents poids s'auto-adaptent comme décrit dans l'algorithme 2, puis chaque situation calcule son nouveau degré d'anomalie en utilisant les poids mis à jour.

À la réception des paramètres envoyés par la situation : le sens, la criticité et l'influence, les agents poids décident de mettre à jour leur poids ou non, indépendamment et simultanément. Pour ce faire, ils suivent l'algorithme 2.

Algorithme 2 Auto-adaptation des agents poids

Entrée: $Sens, Crit, Influence$ de toutes situations

1: Sélectionner la situation normale la plus critique NSA et la situation anormale la plus critique ASA
2: **si** $Crit(NSA) > Crit(ASA)$ **alors**
3: **si** $Influ(NSA) > Influ(ASA)$ **alors**
4: l'agent poids diminue sa valeur
5: **fin si**
6: **si** $Influ(ASA) > Influ(NSA)$ **alors**
7: l'agent poids augmente sa valeur
8: **fin si**
9: **fin si**

Dans le tableau 1, la ligne 5 n'est pas une anomalie même si la valeur de 400 du capteur 5 est très éloignée de la valeur nominale qui est de 65. Elle n'est pas considérée comme une anomalie par l'expert, ce qui signifie que ce capteur est soit sans importance pour la détection des anomalies, soit que la variation n'est pas assez grande pour qu'un expert la considère comme anormale. La ligne 5 n'est pas une anomalie même si la valeur de 400 du capteur 5 est très éloignée de la valeur nominale qui est de 65. Elle n'est pas considérée comme une anomalie par l'expert, ce qui signifie que ce capteur est soit sans importance pour la détection des anomalies, soit que la variation n'est pas assez grande pour qu'un expert la considère comme anormale.

La coopération entre les agents poids garantit qu'au moins un agent poids actualise son poids et que chaque cycle d'actualisation des agents poids se traduit par un meilleur état général où l'agent poids le plus critique diminue sa criticité [16]. Après chaque cycle de mise à jour des agents poids, les situations calculent leur nouvelle criticité en tenant compte des nouvelles valeurs des agents poids. Ce processus est répété jusqu'à ce que les situations les plus critiques étiquetés "normales" et "anormales" aient leurs criticités respectives égales et négatives. Cela garantit que toutes les situations sont correctement classées et que les situations suivantes ont de meilleures chances d'être classées en conséquence par SANDMAN. En effet, la meilleure façon de réduire le taux de faux négatifs et de faux positifs [3], et donc d'avoir un système robuste, est de maintenir les situations connues les plus critiques (sauvegardées dans l'historique) aussi loin du seuil que possible.

4 Expérimentations

Dans cette section, des expérimentations sont menées pour montrer la capacité de SANDMAN à détecter plusieurs types d'anomalies en utilisant les feedbacks d'un expert.

4.1 Description du cadre expérimental

Les données utilisées pour mener à bien les expérimentations ont été générées à l'aide du générateur de séries temporelles TSimulus [17]. Ces séries temporelles comportent une valeur mesurée signée pour chaque capteur pour chaque heure. La plage de valeurs et l'amplitude de variation est spécifique à chaque capteur. Les valeurs pour chaque capteur sont cycliques, avec ou sans bruit sur une période de 24 heures, c'est-à-dire que la valeur d'un capteur à 15 heures est la même chaque jour (au bruit près). Ces données synthétiques sont modifiées par un expert humain afin d'y introduire des anomalies de différents types. SANDMAN traite les données en temps réel et l'expert donne son avis a posteriori de manière asynchrone.

Toutes les expérimentations ont été réalisées sur un processeur à 4 coeurs d'une fréquence de 2,6 GHz. Les données utilisées dans chaque expérimentation ont des mesures toutes les heures pendant un mois, pour un total de 744 heures. 4 fichiers de données distincts sont utilisés pour montrer la capacité de SANDMAN à : i) détecter les anomalies ponctuelles et pallier le bruit, ii) détecter des anomalies collectives et contextuelles, iii) adapter les profils des capteurs, iv) passer à l'échelle.

Dans les expériences, nous avons choisi de reproduire l'ensemble des données d'un mois afin de montrer l'amélioration apportée par l'apprentissage du SANDMAN. Ainsi, les résultats sont présentés après un mois et deux mois.

Les résultats présentés dans cette section sont le résultat de ces deux mois. Les résultats sont présentés sous la forme du nombre de :

— VN : Vrai Négatif, VP : Vrai Positif,
— FN : Faux Négatif, FP : Faux Positif,
— t/sit : temps de calcul par situation traitée en ms.

4.2 Anomalies ponctuelles et résilience au bruit

Les anomalies ponctuelles sont le type d'anomalie le plus souvent détecté dans la littérature, car les approches univa-

riées et multivariées peuvent les détecter.
Dans notre expérimentation, les données mesurées sont bruitées comme le sont les données issus de capteurs vieillissant ou ayant une précision limitée. La base de données contenant des anomalies ponctuelles est utilisée pour étudier l'effet du bruit des données sur la précision de la détection des anomalies et les performances du système. Le fichier de données contient 20 capteurs, soit 20 x 744 = 14880 données. 58 anomalies ponctuelles ont été ajoutées manuellement par un expert. Le bruit est simulé en appliquant une distribution uniforme à chaque capteur avec la formule suivante :

$$Valeur_{bruit} = Valeur_{sansBruit} - \frac{intervalle}{2} + rand_{[0,1]} * intervalle * bruit$$

Avec :

- $intervalle$: $|MIN(c) - MAX(c)|$ pour chaque capteur c
- $bruit$: la quantité de bruit, respectivement 1% et 5% dans les expérimentations.

Le tableau 2 montre qu'un niveau faible de bruit n'a pas d'impact sur le fonctionnement de SANDMAN. Cependant un niveau de bruit plus élevé mène à un plus fort taux de faux positifs. SANDMAN peut réduire ce taux de faux positifs en créant des situation normales et ainsi apprendre des poids plus bas.

4.3 Anomalies collectives et contextuelles

Dans cette expérimentation, la base de données contient des anomalies ponctuelles, contextuelles et collectives. Les anomalies contextuelles surviennent lorsqu'une valeur de capteur est inhabituelle plusieurs fois de suite. Les anomalies collectives surviennent lorsque plusieurs valeurs de capteur pour la même heure sont inhabituelles, mais pas assez pour qu'une seule d'entre elles provoque une anomalie ponctuelle.
Le tableau 3 donne le nombre de chaque type d'anomalie, ainsi que le nombre de vrais positifs et faux négatifs au premier mois, puis au deuxième mois. Il n'y a pas de faux positifs dans les résultats de cette expérimentation. SANDMAN est capable de créer et d'ajouter à l'historique les situations correspondant aux situations mal classifiées au mois 1 pour les classifier sans erreur au mois 2, et ce, quelque soit le type d'anomalie.

4.4 Mise à jour du profil sur des données glissantes

Dans les expérimentations précédentes, le profil de chaque capteur a été stocké et a contribué à la détection des anomalies, mais les données étaient cycliques sur une base journalière. Dans cette expérimentation, les valeurs mesurées des capteurs sont modifiées de manière à ce que chaque valeur mesurée soit supérieure de 1% chaque jour, de sorte que sur un mois de 31 jours, chaque capteur a des valeurs mesurées supérieures de 31% aux données de contrôle. Le fichier de test comporte 58 anomalies ponctuelles et chacun des 20 capteurs est la cause d'une anomalie au moins une fois. Le tableau 4 présente les résultats de l'expérimentation. La mise à jour du profil des capteurs est suffisamment réactive pour permettre à SANDMAN de classifier les anomalies quand les valeurs mesurées des capteurs changent au cours du temps.

	Mois 1				Mois 2			
Bruit	TP	TN	FN	FP	TP	TN	FN	FP
Pas de Bruit	40	686	18	0	58	686	0	0
1%	40	686	18	0	58	686	0	0
5%	58	655	0	31	58	679	0	7

TABLE 2 – Résultats de la détection d'anomalies ponctuelles avec bruit

		Mois 1		Mois 2	
	nb	TP	FN	TP	FN
Ponctuelle	6	2	4	6	0
Collective	5	1	4	5	0
Contextuelle	3	1	2	3	0
Total	14	4	10	14	0

TABLE 3 – Résultats de la détection des 3 types d'anomalies

4.5 Passage à l'échelle

Dans cette expérimentation, un nombre croissant de capteurs est utilisé pour mesurer l'effet d'un nombre de capteurs plus important sur le temps de calcul. Pour ce faire, un ensemble de données sur 20 capteurs distincts est dupliqué pour obtenir jusqu'à 800 capteurs. Les données utilisées sont les mêmes que pour l'expérience précédente sur les données glissantes sans la modification des valeurs de 1% par jour. Le tableau 5 présente le temps de calcul par situation en fonction du nombre de capteurs. Le temps de calcul au mois 2 est toujours plus court que celui du mois 1 car l'apprentissage des poids est effectué uniquement en mois 1. On remarque également que le temps de résolution est proportionnel au nombre de capteurs utilisés et que la différence de temps entre les deux mois est constante. Ceci est dû au fait que l'essentiel du temps d'exécution vient de la lecture des données brutes à partir d'une base de don-

	Mois 1				Mois 2			
	TP	TN	FN	FP	TP	TN	FN	FP
Valeurs glissantes	46	686	12	0	58	686	0	0

TABLE 4 – Résultats de l'adaptation du profil

Nombre de capteurs	Mois 1 t/sit (ms)	Mois 2 t/sit (ms)
20	2.7	1.7
40	4.8	3.5
100	22	20
200	48	45
400	102	100
800	185	180

TABLE 5 – Résultats du passage à l'échelle

nées, qui a un coût fixe par capteur.

5 Conclusions et Perspectives

Concernant la gestion des données liées à l'énergie dans un bâtiment intelligent, plusieurs types d'anomalies doivent être détectées. Elles peuvent être ponctuelles, collectives ou contextuelles. La détection de ces différents types d'anomalies est absente des méthodes de l'état de l'art pour lesquelles le prétraitement des données est obligatoire. Nous avons donc proposé SANDMAN, un système de détection d'anomalies en temps réel semi-supervisé qui utilise des données brutes comme entrée et classe les anomalies en apprenant avec les feedback de l'expert. SANDMAN est capable de détecter les trois types d'anomalies de manière générique et s'adapte bien à un nombre croissant de capteurs. Le feedback de l'expert est réduit au minimum car celui-ci est facultatif et SANDMAN peut fonctionner sans étiquetage préalable. Après réparation d'une anomalie détectée par SANDMAN, il doit automatiquement observer le retour à des valeurs nominales des capteurs pour inhiber l'anomalie en cours. C'est l'amélioration que nous développons actuellement. SANDMAN devra également être capable d'apprendre plusieurs séries de profils pour les capteurs, afin de tenir compte des différents comportements des utilisateurs des bâtiments (par exemple les périodes semaine et week-end).

Références

[1] "Energy performance of buildings." https ://ec.europa.eu/energy/en/topics/energy-efficiency/energy-performance-of-buildings. Accessed : 2019-11-7.

[2] J. I. Guerrero, A. García, E. Personal, J. Luque, and C. León, "Heterogeneous data source integration for smart grid ecosystems based on metadata mining," *Expert Systems with Applications*, vol. 79, pp. 254–268, 2017.

[3] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection : A Survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, 2009.

[4] T. Labeodan, K. Aduda, G. Boxem, and W. Zeiler, "On the application of multi-agent systems in buildings for improved building operations, performance and smart grid interaction – A survey," *Renewable and Sustainable Energy Reviews*, vol. 50, 2015.

[5] W. Turner, A. Staino, and B. Basu, "Residential HVAC fault detection using a system identification approach," *Energy and Buildings*, vol. 151, pp. 1–17, 2017.

[6] C. Miller, Z. Nagy, and A. Schlueter, "A review of unsupervised statistical learning and visual analytics techniques applied to performance analysis of non-residential buildings," *Renewable and Sustainable Energy Reviews*, vol. 81, pp. 1365–1377, 2018.

[7] K. Yan, Z. Ji, and W. Shen, "Online fault detection methods for chillers combining extended kalman filter and recursive one-class SVM," *Neurocomputing*, vol. 228, pp. 205–212, 2017.

[8] J.-S. Chou and A. S. Telaga, "Real-time detection of anomalous power consumption," *Renewable and Sustainable Energy Reviews*, vol. 33, pp. 400–411, 2014.

[9] J. Wu, W. Zeng, and F. Yan, "Hierarchical Temporal Memory method for time-series-based anomaly detection," *Neurocomputing*, vol. 273, pp. 535–546, 2018.

[10] Y. Zhu, X. Jin, and Z. Du, "Fault diagnosis for sensors in air handling unit based on neural network preprocessed by wavelet and fractal," *Energy and Buildings*, vol. 44, pp. 7–16, jan 2012.

[11] A. Capozzoli, F. Lauro, and I. Khan, "Fault detection analysis using data mining techniques for a cluster of smart office buildings," *Expert Systems with Applications*, vol. 42, no. 9, pp. 4324–4338, 2015.

[12] M. Peña, F. Biscarri, J. I. Guerrero, I. Monedero, and C. León, "Rule-based system to detect energy efficiency anomalies in smart buildings, a data mining approach," *Expert Systems with Applications*, vol. 56, pp. 242–255, 2016.

[13] P. Xue, Z. Zhou, X. Fang, X. Chen, L. Liu, Y. Liu, and J. Liu, "Fault detection and operation optimization in district heating substations based on data mining techniques," *Applied Energy*, vol. 205, pp. 926–940, 2017.

[14] A. Forestiero, "Self-organizing anomaly detection in data streams," *Information Sciences*, vol. 373, pp. 321–336, 2016.

[15] Y. Seng Ng and R. Srinivasan, "Multi-agent based collaborative fault detection and identification in chemical processes," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 6, pp. 934–949, 2010.

[16] J.-P. Georgé, M.-P. Gleizes, and V. Camps, "Cooperation," in *Self-organising Software* (G. Di Marzo Serugendo, M.-P. Gleizes, and A. Karageorgos, eds.), Natural Computing Series, pp. 193–226, Springer, 2011.

[17] "Tsimulus." https ://tsimulus.readthedocs.io/en/latest/. Accessed : 2019-12-18.