POLYPUBLIE
Polytechnique Montréal

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Titre: Title:	Analyse et détection des anomalies dans un système IOT distribué hybride
Auteur: Author:	Cyrine Zid
Date:	2020
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Zid, C. (2020). Analyse et détection des anomalies dans un système IOT distribué hybride [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/5403/

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/5403/
Directeurs de recherche: Advisors:	Giuliano Antoniol, & Gabriela Nicolescu
Programme: Program:	Génie informatique

Ce fichier a été téléchargé à partir de PolyPublie, le dépôt institutionnel de Polytechnique Montréal
This file has been downloaded from PolyPublie, the institutional repository of Polytechnique Montréal

https://publications.polymtl.ca

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Analyse et détection des anomalies dans un système IOT distribué hybride

CYRINE ZID
Département de génie informatique et génie logiciel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie informatique

Août 2020

© Cyrine Zid, 2020.

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Analyse et détection des anomalies dans un système IOT distribué hybride

présenté par **Cyrine ZID**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Jinghui CHENG, président
Giuliano ANTONIOL, membre et directeur de recherche
Gabriela NICOLESCU, membre et codirectrice de recherche
Marios-Eleftherios FOKAEFS, membre

DÉDICACE

Je dédie ce travail à mes parents, ma soeur, ma famille, mes amis et à toute personne qui m'a soutenue et a cru en moi

Merci...

REMERCIEMENTS

En premier, je tiens à remercier profondément mon directeur de recherche professeur ***Antoniol Giuliano*** pour sa disponibilité, son encadrement et ses commentaires constructifs, tout au long de la réalisation de ce projet. Je profite de l'occasion pour lui témoigner ma reconnaissance et ma profonde gratitude pour la confiance placée en moi et pour le soutien scientifique, moral et financier qu'il m'accorde. Travailler au côté d'une personne aussi passionnée est une chance inespérée dont seuls certains jouissent.

Je tiens aussi à remercier ma codirectrice de recherche professeure ***Nicolescu Gabriela*** pour ses précieux conseils et son accompagnement dans des choix parfois difficiles et cela tout le long de mes deux années de maîtrise.

Je remercie aussi le professeur ***Foutse Khomh*** pour son expertise et son aide lors de la réalisation de ce projet.

Je tiens, également, à remercier mes parents qui ont cru en moi, depuis mon plus jeune âge, et m'ont toujours soutenu dans mes projets de vie. Sans vous, je ne serais jamais devenue la personne que je suis aujourd'hui et je vous en suis profondément reconnaissante.

Je remercie aussi ma petite sœur pour son encouragement et le bonheur qu'elle me procure constamment.

Je remercie mes tantes pour l'amour, les conseils et leur soutien inestimable.

Je remercie mes amis pour leur soutien moral constant. Grâce à vous, je me sens chez moi même à des milliers de kilomètres. Vous êtes ma deuxième famille.

Enfin, le présent étant la suite du passé, je tiens à remercier tous les professeurs ayant participé à la collecte du savoir qui m'a menée jusqu'ici.

RÉSUMÉ

Un système domotique intègre divers appareils intelligents (c'est-à-dire des dispositifs cyber-physiques), des dispositifs virtuels et des applications logicielles, cela, dans le but de contrôler, surveiller et gérer une ou plusieurs maisons.

Les dispositifs, applications et systèmes intelligents offrent d'innombrables possibilités, mais leur nature de systèmes cyberphysiques interconnectés, hétérogènes, distribués et orientés vers le consommateur pose de nouveaux défis aux utilisateurs , aux développeurs et aux responsables de la maintenance. Dans ce mémoire, nous nous concentrons sur les systèmes domotiques construits avec des dispositifs autour d'un protocole qui ne dépend pas du fournisseur, à savoir Z-Wave. La disposition, les composants et les applications logicielles de l'IOT évoluent et changent, au fil du temps, et cela en fonction de l'évolution des besoins des utilisateurs. Malgré les améliorations constantes, la détection des défauts et la maintenance du système restent assez difficiles. Il n'est pas rare de voir des actions manquantes ou des événements inattendus.

Généralement, les systèmes domotiques sont construits et déployés sur une infrastructure existante par les utilisateurs eux-mêmes. Les utilisateurs peuvent ne pas être très à l'afflux des technologies ou ne pas avoir une bonne compréhension des réseaux, des protocoles et de l'architecture des systèmes. Les solutions sont construites une pièce à la fois. Les fonctionnalités et les composants sont ajoutées et évoluent progressivement avec des risques de non-considération pour l'interopérabilité, les dépendances, les interférences des fonctionnalités ou les problèmes de maintenance futurs.

Dans ce mémoire, nous proposons des approches pour tracer les anomalies dépendantes du temps dans les systèmes cyberphysiques grâce a des mécanismes non supervisés pour signaler la présence de problèmes. Les résultats montrent que différents types d'anomalies peuvent être détectés dans un système "smart-home" en utilisant des entrées de logs, des données de sniffage radio, des analyses d'intervalles et de séries temporelles. L'approche est suffisamment générale pour pouvoir être adaptée à un large éventail de réseaux IOT afin d'identifier et de suivre les anomalies et les comportements indésirables. Cela dans un but ultime d'alerter l'utilisateur et de rendre le système plus robuste.

ABSTRACT

A home automation systems integrates various smart devices (*i.e.,* cyber-physical devices), virtual devices, software application with the goal to control, monitor and manage one or more houses.

Smart devices, applications, and systems offer new countless possibilities, however, their nature of consumer oriented, distributed, heterogeneous interconnected cyber-physical systems poses new challenges to users, developers and maintainers. In this thesis we focus on home automation systems built with devices around a vendor agnostic protocol, namely, Z-Wave. Layout, components and software applications for IOT evolve and change over the time with ever changing user need. Despite constant improvements, defect detection and system maintenance remain quite challenging. It is not uncommon to see missing actions or unexpected events.

Often, home automation systems are built and deployed over an existing infrastructure by the users themselves. Users may not be tech-savvy or have a strong understanding of networking, protocols and system architecture. Solutions are built one piece at a time; functionalities and components are added and evolved gradually with risks of little or no consideration for interoperability, dependencies, features interferences or future maintenance issues.

In this thesis we propose approaches to trace time dependent anomalies in cyber-physical systems.

The goal is to have unsupervised mechanisms to alert the presence of problems and possibly fix them. Results show that various types of anomalies can be detected in a smart-home system using log entries, radio sniffing data, time interval and time series analysis. The approach is general enough to be adaptable to a wide range of IOT networks to identify and track anomalies and unwanted behavior with the ultimate goal to alert the user and make the system more robust.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

IOT	Internet Of Things
OSI	Open Systems Interconnection
MQTT	Message Queuing Telemetry Transport
OASIS	Organisation For The Advancement Of Structured Information Standards
ST	Série Temporelle

CHAPITRE 1 INTRODUCTION

1.1 Contexte

Depuis plus d'un demi-siècle, les chercheurs et les industriels investissent dans le progrès de la microélectronique. Ils adoptent le chemin de prédiction de Gordon Moore qui affirme que : "le nombre de transistors présents dans un microprocesseur continuerait de doubler tous les 18 mois". Cette règle nous permet d'observer, non seulement, une évolution progressive de ces composants électroniques, mais aussi de l'innovation dans de nombreux secteurs industriels.

Ainsi, grâce aux progrès de la technologie de fabrication moderne, de nouveaux produits électroniques tels que les appareils mobiles intelligents, les systèmes intelligents autonomes. peuvent, désormais, être construits à **faible coût**. Cette évolution a, également, rendu possible le développement des télécommunications et des réseaux. En effet, sans électronique complexe, il n'y aurait pas de centraux téléphoniques modernes, pas d'informatique distribuée, pas d'internet, pas de téléphone mobile.

L'internet des objets "IOT" est une technologie de l'information et de la communication distribuée qui intègre plusieurs objets, physiques ou virtuels. Ils peuvent être répartis sur plusieurs zones géographiques. Ils sont capables de collecter et de transférer des données sur des systèmes connectés de façons autonomes et fiables afin d'accomplir de nombreuses tâches.

IOT est une technologie cyber-physique. Un système cyber-physique est un mécanisme autonome capable d'échanger des informations dans l'environnement ou il se trouve et de déclencher des actions. Il intègre, également, des composants matériels, logiciels et des éléments de communication afin de surveiller et agir en temps réel sur le monde physique [2]. Cette particularité lui permet d'être implanté dans plusieurs domaines d'utilisation parmi les suivants (figure 1.1) :

- santé
- environnement
- infrastructure/habitation
- service public
- sécurité

Comme mentionné précédemment, un IOT est un ensemble de dispositifs et d'applications qui offrent d'innombrables possibilités. Toutefois, la nature de ce système cyber-physique c'est-à-dire distribué, hétérogène et inter-connecté, pose de nouveaux défis aux utilisateurs, agents de maintenance et développeurs. Même si ce type de systèmes simplifie la vie, ils sont

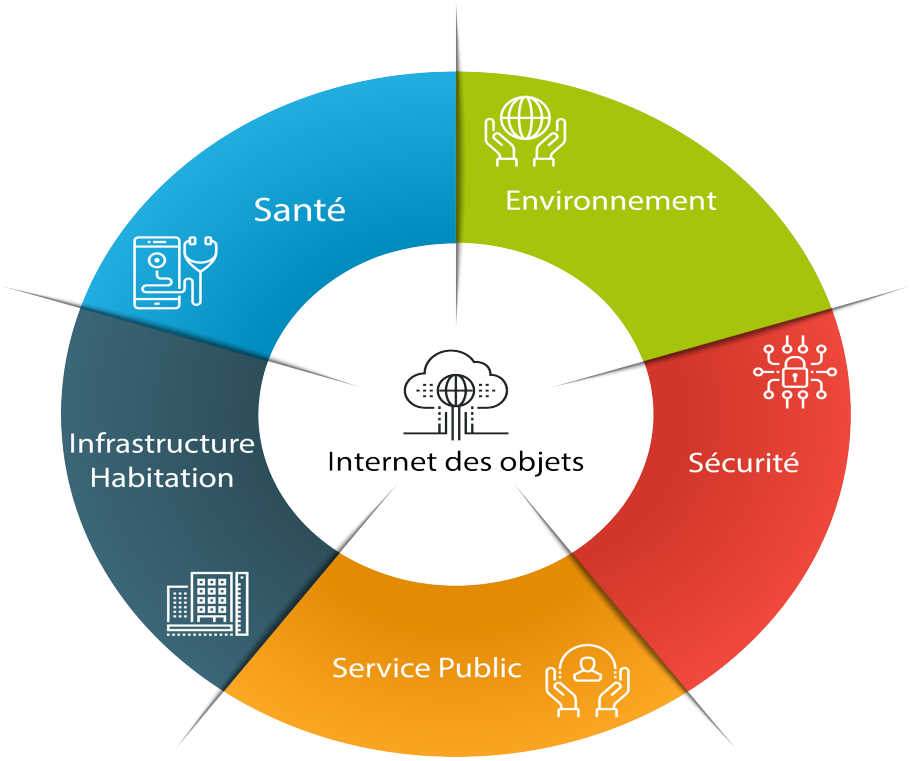


Figure 1.1 Architecture général d'un système IOT

souvent des boîtes noires, complexes à déboguer et à maintenir.

La figure 1.2 représente une architecture simplifiée d'un système IOT où nous retrouvons trois couches principales. Ces couches représentent [1] :

- la perception : des compassant matériels qui interagissent avec l'environnement réel (collecte de données, action, notification).
- le réseau : une passerelle entre les capteurs implantées sur le réseau et la couche application.
- l'application : stockage, traitement et analyse des données reçu. Elle donne aussi la main à l'utilisateur de personnaliser quelques fonctionnalités.

Tout dépendra, alors, de l'architecture que nous voulons implémenter. Nous pouvons, par exemple, disposer d'une couche métier et/ou d'une couche traitement. Cela ne sera que le résultat d'une conception préétablie.

Le domaine qui nous intéresse le plus est celui de l'habitation, plus communément appelée la domotique. Un système domotique offre des solutions regroupant les techniques permettant de contrôler, d'automatiser et de programmer l'habitat *e.g.,* nous pouvons avoir un contrôleur

1. https ://www.netburner.com/learn/architectural-frameworks-in-the-iot-civilization/

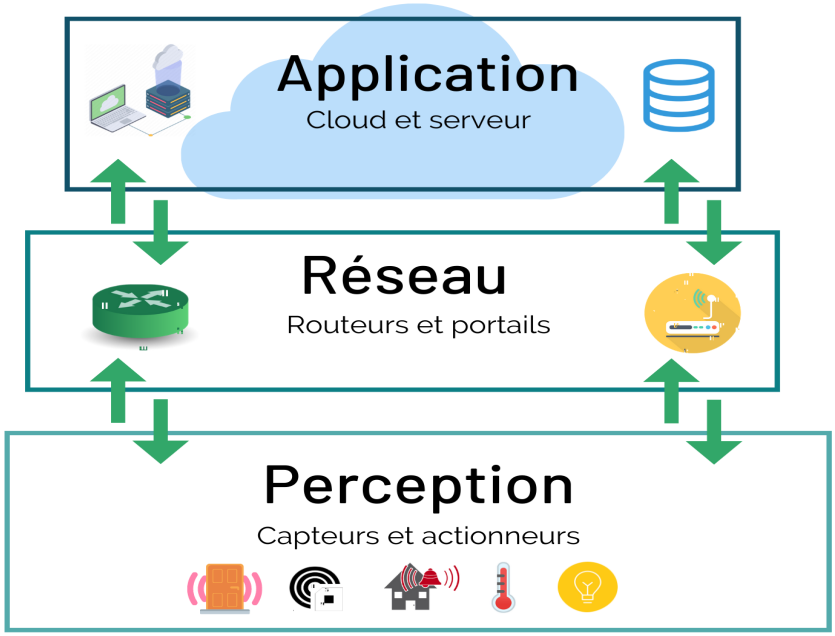


Figure 1.2 Architecture général d'un système IOT

capable de manager tous les composants d'un immeuble ou d'une maison (le contrôle des ascenseurs, de la climatisation)

De manière générale, il existe deux principales catégories de dispositifs domotiques. Nous retrouvons, d'une part, les produits verrouillés tels Google Home ou Amazon Alexa dont l'accès à l'information est restreint ; les informations de journal y sont rares et les protocoles de communication non-divulgués. D'autre part, il existe des dispositifs de produits non-verrouillés où l'accès aux détails est envisageable. Nous pouvons, donc, légitimement, nous interroger sur la manière dont il convient de choisir les dispositifs, en fonction des attentes du consommateur, les exigences de confidentialité ou l'expertise de l'utilisateur final.

Tout d'abord, notons que nous allons travailler sur un modèle déployé intégrant plusieurs composants et dispositifs ; sur ces composants physiques nous avons plusieurs couches logiciels développées au fil des années. Il est implanté de façon à satisfaire les exigences de notre utilisateur, néanmoins, ce genre de système est modulable et offre plusieurs scénarios qui s'adaptent au besoin de chacun.

Prenons, par exemple, comme appareils, deux thermostats servant à contrôler, à distance, la température de la maison. Ensuite, ajoutons leurs d'autres dispositifs aidant à contrôler

la climatisation et/ou l'éclairage. Ainsi, à chaque étape, de nouvelles applications logicielles sont greffées et de plus en plus de données sont générées. Nous pouvons en conclure que ce processus est incrémental et peu contrôlé. La fiabilité et l'interopérabilité du matériel et des logiciels ne sont pas garanties. Par ailleurs, des risques d'interactions entre différentes fonctionnalités peuvent survenir augmentant ainsi le caractère imprédictible du processus et engendrant une mauvaise maîtrise du système.

En conclusion, vu que la complexité du système augmente, en ajoutant règles logiques, scénarios, composants matériels et logiciels, les défis que rencontrent le développeur sont en constante augmentation.

1.2 Problématique

Une série chronologique nous permet d'analyser, de décrire, d'expliquer et de prévoir l'évolution d'un phénomène au cours du temps. Elle peut présenter une tendance, un aléa (bruit) ou bien un phénomène saisonnier, cyclique ou accidentel. L'internet des objets est une infrastructure de réseau dynamique dans laquelle des sous-systèmes et entités physiques et virtuelles identifiables, autonomes et auto-configurables communiquent entre eux, et qui interagissent avec l'environnement. Les actions produites par ces dispositifs sont déclenchées (et archivées chronologiquement) et il est possible de les modéliser et représenter sous forme de séries temporelles. Ces dernières représentent des séquences de mesure qui, dans le temps, décrivent le comportement de notre système. Nous allons, donc, essayer de les analyser et interpréter afin d'y déceler les éventuelles anomalies que nous définirons par la suite.

La détection d'anomalies dans des systèmes en temps réel peut être effectuée à partir de l'interprétation d'une analyse séquentielle de temps. Dans ce projet, le système implanté génère une quantité importante d'informations : la date, le temps, le message. Dans ce contexte, une anomalie se définit comme une donnée et/ou valeur inhabituelle ou non-conforme à un résultat attendu. En effet, si notre système est programmé à maintenir une température moyenne de 17 degrés Celsius et qu'à plusieurs reprises nous observons une température de 21 degrés Celsius alors nous pourrions en déduire que nous sommes en présence d'un comportement inattendu, qualifié d'anomalie. Mais, évidement pendant l'été un tel comportement serait tout a fait normal. Notre méthode d'analyse devra, de ce fait, être capable de détecter n'importe quel comportement anormal de façon autonome mais tout en considérant le contexte et les contraintes environnementales.

1.3 Objectifs

1.3.1 Question de recherche

Dans ce projet, nous travaillons sur un système domotique que nous classifierons d'automne, se basant sur un contrôleur central, qui gère tous les différents composants (matériels/logiciels). Ce système utilise le protocole Z-Wave pour la communication entre ces différents capteurs/actionneurs [3]. Il intègre aussi des composants wifi qui fonctionnent en réseau infonuagique, et communiquent avec le système Z-Wave avec le protocole MQTT. Ce mémoire répond à la question de recherche suivante :

En se basant sur les séries temporelles, comment implémenter un mécanisme supervisé, semi-supervisé ou non supervisé (en fonction des contraintes de chaque dispositif et l'environnement) capable de détecter et de signaler une anomalie.

1.3.2 Objectif général

L'objectif général de ce mémoire est d'implémenter une méthode d'analyse de séries temporelles afin d'extraire d'éventuelles anomalies, et cela en se basant sur des méthodes statistiques existantes. Pour ce faire, il nous faudrait d'abord réaliser quelques tâches plus spécifiques que nous allons détailler dans les objectifs spécifiques.

1.3.3 Objectifs spécifiques

Les objectifs spécifiques de ce projet se présentent comme suit :

- Simuler un mini système test dans notre laboratoire.
- Analyser et traiter les données générées par notre mini système afin de les rendre exploitables.
- Observer et comprendre des erreurs non-détectées par le système.
- Implémenter des différentes techniques de prise de décision sur le système réelle.

1.3.4 Résultat

Dans le cadre de notre projet de mémoire, nous avons développé une méthode de détection qui nous a permis d'identifier et définir plusieurs types d'anomalies (causales/fantômes). Nous les avons, ensuite, classifié (continue/discrète). Le cheminement de ces résultats sera expliqué plus en détails tout au long des différents chapitres. L'une des extensions futures

sera de synthétiser des algorithmes qui répareront les anomalies sans intervention extérieure (de l'utilisateur).

1.4 Plan

Le mémoire débutera par la revue de littérature qui explicitera les différents travaux reliés à notre sujet, de même que ceux qui nous ont incité à nous approfondir sur ce genre de problèmes. Le chapitre suivant présentera les connaissances préalables requises pour une bonne compréhension du fonctionnement du système. Une fois tous les concepts définis, le quatrième chapitre présentera l'approche suivie pour atteindre nos objectifs. Nous présenterons, ensuite, notre système-test plus en détails avant de présenter le dernier chapitre qui illustrera les tests réalisés, ainsi que les réponses des différentes questions émises. Le mémoire se terminera avec une conclusion et nos perspectives de projets.

CHAPITRE 2 REVUE DE LITTÉRATURE

La détection d'anomalies dans des séries temporelles est un sujet largement abordé dans la littérature et nous la trouvons dans différents domaines d'utilisation. Dans ce chapitre, nous présenterons les travaux les plus importants en lien avec notre problématique. Nous sommes particulièrement intéressés par la détection des anomalies, les séries temporelles, la sécurité des systèmes connectés, les système IOT et réseau maillé.

2.1 Détection des anomalies et séries temporelles

L'idée de la détection des anomalies/erreurs et de la surveillance des réseaux n'est pas nouvelle. En effet, différentes techniques ont été appliquées au fil des années dépendamment du domaine et/ou problématique. Pour exemple, la détection des anomalies (logicielles et matérielles) constitue, manifestement, l'une des composantes clés du paradigme Industry 4.0 [4]. Elle s'applique également aux techniques bien connues "condition-based maintenance (CBM)" et "prognosis and health management (PHM)" [5]. Toutefois, ces approches sont, principalement, axées sur le traitement des données provenant des équipements industriels.

Dans l'ouvrage [6], les auteurs proposent un nouvel outil d'analyse "failure mode and effects analysis (FMEA) ". Ils étudient le niveau de bruit des réfrigérateurs qui ont montré des problèmes de maintenance et, en prenant les données historiques des capteurs de bruit sans fil correspondants comme données d'entraînement, construisent un modèle de régression logistique pour la prédiction de la probabilité de défaillance.

Pour les systèmes IOT ou appartenant à l'industrie 4.0, un volet important est couvert par les problèmes liés spécifiquement au réseau sans fil. En effet, des méthodes plus spécifiques pour l'identification des problèmes dans les réseaux sans fil, sont présentées dans [7, 8]. Les auteurs mentionnent que les défauts sont inévitables et que les réseaux de capteurs sans fil sont susceptibles d'être défectueux, ce qui peut être dû à un comportement anormal des logiciels ou du matériel ou à une mauvaise qualité des liaisons de communication. Pour appuyer la déclaration précédente, Tolle et al. [9] indiquent dans leur étude que seulement 49% des données collectées pourraient être utilisées pour une interprétation significative dans un réseau de capteurs pour l'examen des microclimats de la flore.
Il est donc clair que des techniques doivent être développées pour détecter et prévenir les anomalies dans les réseaux d'IOT. Leur majeure partie est constituée d'environnements intelligents, distribués et interconnectés selon diffèrent mécanismes (avec ou sans fil).

Plus en général, la détection des anomalies est depuis longtemps un domaine de recherche important : voir l'étude [10] ; des méthodes de séries temporelles univariées [11], des séries temporelles multi-variées [12], des approches graphiques [13], et des framework ont été proposés [14] pour la détection des anomalies. Récemment, des techniques d'apprentissage approfondies ont également été appliquées à la détection des anomalies [15].
Compte tenu de la quantité de données disponible et de la nature exploratoire de notre travail, nous avons adopté des techniques plus traditionnelles basées, soit sur décision statistique, soit sur une certaine mesure de dissimilitude. En effet, les concepts de similarité, de distance et de divergence sont au cœur des différentes analyses. Une question problématique dans l'IOT et plus généralement dans les systèmes cyberphysiques (SCP), est que pour un intervalle de temps donné, différents dispositifs peuvent produire un nombre différent d'observations. Ainsi, même s'il est utile d'envisager des analyses où les fenêtres temporelles contiennent un nombre fixe d'échantillons (*e.g.,* voir [11]), il est plus naturel de comparer des sous-séquences contenant un nombre différent de mesures.

Par exemple, deux capteurs de température signalant les changements de température ambiante, pendant une heure, et cela avec une précision de 0,5 degré Celsius, généreront probablement un nombre différent d'échantillons selon la façon dont la température des pièces a fluctué. Pour faire face à des fenêtres temporelles de longueur variable, nous avons recours à un outil traditionnel d'analyse des séries temporelles, à savoir le Dynamic Time Warping (DTW) [16,17]. *Notre objectif est d'adopter et d'adapter des approches et des tests statistiques développés antérieurement afin d'exploiter différents types d'anomalies.* De manière générale, nous visons à détecter l'instant où une ou plusieurs variables mesurées s'écartent de manière significative du comportement normal attendu sur la base de données historique [12].

Comme le suggèrent Cheng et al. [12], les anomalies peuvent, également, être classées comme des anomalies de type *discret* et *continu*. En quelques mots, par exemple, une anomalie de séquence pour une fenêtre temporelle donnée sur des échantillons N, peut être détectée, par force brute, en comparant toutes les fenêtres non croisées par rapport à une distance [11] et en sélectionnant les segments les plus discordants. Par contra, une technique de détection d'anomalies ponctuelles cherche à trouver le moment où les valeurs diffèrent des données historiques, c'est-à-dire le reste de la série temporelle. Au contraire, une anomalie de séquence est un segment de la série temporelle qui n'a pas d'équivalent dans la série temporelle et qui est composé uniquement de valeurs aberrantes.

Plus précisément concernant les séries temporelles, une anomalie peut être locale ou globale, selon la définition du comportement normal. Dans une anomalie locale, le comportement est défini par rapport à un intervalle contenant la datation de l'anomalie, alors qu'une anomalie

globale considère la série temporelle entière. Un type spécifique d'anomalies ponctuelles est constitué de valeurs aberrantes ; celles-ci peuvent être détectées par diverses techniques et heuristiques.

2.2 Sécurité des systèmes connectés

L'une des caractéristique majeures des systèmes domotiques est l'interconnectivité qui est, à la fois, sa plus grande force et sa plus grande faiblesse [18–20]. La problématique majeure qui en découle est la sécurité et le maintien de la confidentialité des données qu'ils génèrent.

D'après l'article [21] les dispositifs dans les système IOT fonctionnent à faible puissance. De ce fait, ils disposent d'une mémoire très limitée et de radios à faible consommation d'énergie. De plus, ils se doivent d'accomplir plusieurs tâches de sécurité telles que : l'authentification, l'autorisation, la confidentialité et l'intégrité des données. Il ne faut pas oublier que ces mêmes dispositifs sont exposés à des risques physiques tels que l'humidité, la chaleur. Cette étude explique, également, que les systèmes IOT sont vulnérables aux attaques connues qui sont, en général, observées sur n'importe quel système informatique. L'on note par exemple : le déni de service distribué ou l'attaque humaine, l'attaque de la map ou du routeur, l'attaque par un tiers et l'attaque de rang.

Nous avons remarqué que plusieurs articles alertent sur le fait que, même si les maisons intelligentes vendent du rêve, elles continuent de représenter un réel danger que certaines personnes minimisent ou n'en ont pas conscience [18–20]. Plusieurs problématiques sont évidents [1] ainsi que des solution partielles [2]. En effet, un système distribuée est généralement constitué de plusieurs couches, chaque couche pouvant avoir ses faiblesses.

Avant même d'aborder le sujet des attaques extérieures, nous commencerons par rappeler la présence d'attaques intérieures. Nous prendrons, alors, pour exemple, le scandale d'Alexa, l'assistant personnel virtuel développé par Amazone, qui enregistre indéfiniment ses interactions avec les utilisateurs pour les transmettre à des tiers par la suite. [3]

Nous pouvons, également, prendre pour exemple le scénario d'un hacker qui essaierait de s'attaquer directement au composant physique, ou bien, à l'application de contrôle du système [19,20]. Est-ce que le fournisseur nous garantirait-t-il une protection de ce genre d'intrusion ? Cela reste évidemment une question rhétorique. Cet article vous présentera toutes les façons

1. `https://www.tomsguide.com/news/billions-of-smart-home-devices-open-to-attack-what-to-do`
2. `https://www.iotsecurityfoundation.org/how-to-protect-connected-home-devices-and-appliances-from-cyber-attacks/`
3. https ://ici.radio-canada.ca/nouvelle/1208985/amazon-alexa-donnees-personnelles-enregistrements-audio-transcriptions

possibles de hacker un système. Ce dernier étant apparu en en juillet 2019, un an après la déclaration de Z-Wave concernant le renfoncement de sécurité de leur système [4]. Voici un second article qui appuiera l'idée que c'est possible de rentrer dans un système [5] [6].

2.3 IOT et Réseau maillé

L'internet des objets IOT révolutionne le monde en permettant la collecte automatique de données et leurs utilisations par le biais des communications de machine à machine (M2M). Les applications sont nombreuses [22–24], ainsi que les plateformes et les solutions [25, 26].

La topologie maillée est, souvent, présente dans les systèmes IOT vu que dans ce type de réseaux chaque nœud peut agir comme un router [3,27,28]. De plus, l'une de ces particularités n'est autre que l'adaptation du réseau, lors de l'inclusion/l'exclusion d'un nœud, sans gestion particulière et une autonomie. Ce qui justifie sa robustesse. En effet, une panne d'un nœud ne déclenchera pas une panne générale dans le système.

Même si en théorie nous avons du mal à détecter les failles en vrai, il faut se dire que chaque nœud va avoir deux rôles cruciaux, dont celui de passerelle et en même temps, il doit envoyer ses propres messages.
Le fait que nous pouvons avoir plusieurs passerelles sur un seul router/réseau peut avoir un problème de latence d'envoi de message en général et cela avec problèmes de perte de message ou timeout.

Dans l'univers IOT, la consommation d'énergie est aussi un problème [29], il faut que les dispositifs possèdent une certaine puissance de transmission et une vitesse acceptable. Cette problématique peut amener à une forte consommation d'énergie ou à une vitesse à basse et haute latence avec des problématique de delay ou perte de données. Pour palier à ces problèmes, plusieurs approches sont possible [30, 31].

Notre démarche a une perspective différente des travaux décrits ci-dessus. Nous voulons vérifier de façon automatique et en un temps records, la présence d'anomalies, identifier le jour, l'heure et les composants impliqués sans une grande complexité de calcul. Fondamentalement, notre approche est inspirée du génie logiciel et de la fouille de données. Notre vision est pratique et vise à accompagner les développeurs et les utilisateurs dans l'identification des problèmes existants ainsi que les anomalies conséquentes..

4. https://securelist.com/fibaro-smart-home/91416/
5. https://www.pentestpartners.com/security-blog/z-shave-exploiting-z-wave-downgrade-attacks/
6. https://www.forbes.com/sites/thomasbrewster/2018/05/24/z-wave-hack-threatens-to-expose-100-million-smart-homes/#20a036184517

2.4 Étude comparative des méthodes de détection d'anomalies

Dans l'article [1]"Étude comparative des méthodes de détection d'anomalies" c'est une étude comparative entre 10 revues littéraires où ils ont procédé à une étude de l'art pour l'identification des différentes techniques de détection d'anomalie, les jeux de données et les domaines d'application.

Tableau 2.1 "Synthèse de 10 revues existantes : 1- Hodge et Austin (2004) 2- Patcha et Park (2007) 3- Chandola et al. (2009) 4- Aggarwal (2017) 5-Gupta et al. (2014) 6- Souiden et al. (2016) 7-Tellis et D'Souza (2018) 8-Salehi et Rashidi (2018) 9- Zhang (2013) 10-Chalapathy et Chawla (2019)."extrait de l'article [1]

		1	2	3	4	5	6	7	8	9	10
Techniques	Statistique	×	×	×	×	×	×	×	×	×	
	Clustering	×	×	×	×	×	×	×	×	×	
	Plus proches	×	×	×	×	×	×	×	×	×	
	Classification	×	×	×	×		×				
	Régression				×					×	
	Approche spectrale			×	×						
	Motifs fréquents				×		×				
	Deep learning				×						×
Type de jeux de données	Flux de données				×	×	×	×	×	×	
	Séries temporelles				×	×					×
	Graphes					×			×		
	Grande dimension				×	×			×		
	Séquentielles				×						
	Spatio-temporelles				×	×					×
	Spatiales				×						
Domaines d'application	Détection d'intrusion		×	×	×	×					×
	Détection de fraude			×	×						×
	Santé			×	×	×					×
	Maintenance prédictive			×	×	×					×
	Réseaux de capteurs			×	×	×				×	×
	Traitement d'images			×	×						×
	Traitement de texte			×	×						
	Données biologiques					×					
	Astronomie					×					
	Économie					×					

Dans ce mémoire, nous nous sommes concentrés sur l'analyse des séries temporelles pour la détection de différents types d'anomalie pour cela nous nous sommes inspiré des travaux effectuer.

Le tableau 2.1 nous montres que plusieurs analyses sur les séries temporelles étaient effectuées grâce à différentes techniques : statistiques (paramétriques ou non-paramétriques), basées sur la proximité (le clustering, plus proche voisin) et basées sur le deep learning (algorithmes d'apprentissage automatique supervisé ou non supervisé).

Elles ont été appliquées dans plusieurs domaines tels que la détection d'intrusion et la détection de fraude. Nous allons essayer d'appliquer le même principe, mais dans la détection et la classification d'anomalie.

CHAPITRE 3 CONNAISSANCES PRÉALABLES

Dans ce chapitre nous, présentons tous les pré-requis et les notions essentielles pour comprendre le fonctionnement de notre système et avoir une idée plus claire sur l'objectif que nous voulons atteindre dans notre document.

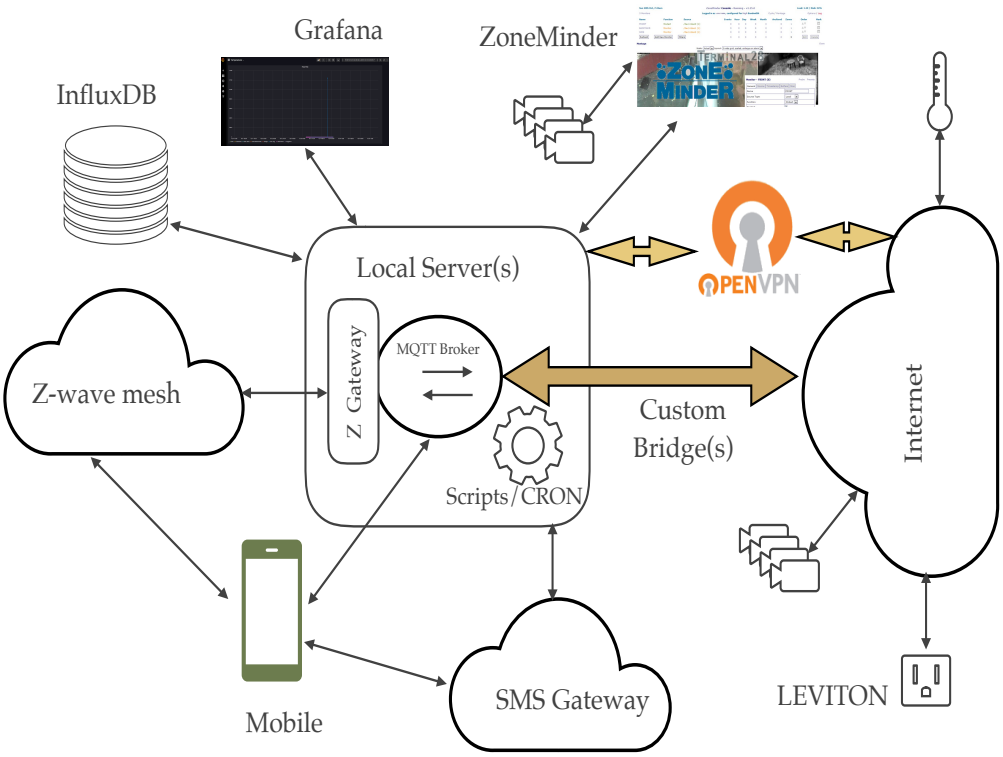


Figure 3.1 Vue d'ensemble du système

Fig. 3.1,représente une vue d'ensemble des différents composants intégrés dans notre système d'essai. Dans les sous-sections suivantes, nous présentons les composantes et les détails essentiels, sans ordre particulier, avec l'objectif de comprendre facilement les choix et les problèmes. L'accent est mis sur l'aspect éminemment pratique, de sorte que, chaque fois que cela est possible, les détails du matériel sont également ajoutés. Certains détails ne peuvent pas être divulgués (*e.g.,* localisation et IP) pour les résonances de confidentialité et de sécurité.

Notre banc d'essai est un prototype d'une maison intelligente : un système hétérogène distri-

bué, où les différents composants interagissent via différents protocoles : TCP (connexion), UDP (protocole de datagramme) et le protocole Z-WAVE.

Au niveau d'abstraction élevée, le protocole Z-Wave est similaire au TCP ; mais les applications sont chargées de gérer l'acquittement des données transmises. En tant que protocole IP, il souffre de trois problèmes principaux :

— La latence
— La perte de paquets
— Le débit

Pour suivre les anomalies dépendantes du temps ou autre évènements indésirables dans une maison intelligente, la première idée serait d'effectuer une extraction de caractéristiques à partir des fichiers journaux [32]. Cependant, il y a plusieurs problèmes. Premièrement, il arrive que les fichiers journaux ne soient pas disponibles comme c'est le cas pour de nombreux dispositifs centrés sur l'infonuagique ; deuxièmement, si une donnée est perdue, il se peut qu'il n'y ait pas de trace dans les journaux et la seule façon possible de découvrir le problème serait de déduire une action causale manquée dans une partie du système.

D'autre part, lorsque les fichiers logs sont disponibles, le nombre de lignes dans ces derniers peuvent facilement atteindre plus d'un million par jour. En effet, le sous-système à Z-Wave de la figure 3.1 produit, en moyenne, un demi-million de lignes de logs par jour ; En plus de ces lignes, il y a les logs générés par le sous-système de vidéo-surveillance ZoneMinder.

Enfin, si le paquet est envoyé et n'est jamais livré, nous pouvons nous tromper en disant que rien de problématique ne s'était produit, alors que c'est le contraire qui est vrai. Ce dernier est en effet un nouveau type de défaillance, très difficile à détecter lorsque deux événements ont une relation de cause à effet.

Dans la fig. 3.2 considérons un premier événement, disons, A, et supposons qu'il se trouve dans le log. Supposons en outre qu'un deuxième événement B, déclenché par A, se trouve également dans le journal. Si B correspond à l'activation d'un actionneur, *e.g.,* à la désactivation d'une vanne, nous pouvons supposer que la vanne a changé de statut. Malheureusement, comme nous l'avons vécu, cela peut être vrai ou non. En effet, si nous considérons les lignes du log dans la figure **??**, même horaire, même dispositifs nous trouvons trois comportements différents.

3.1 Emplacements géographiques et réseaux

Le système s'étend sur quatre sites ; dans chaque site, la connectivité est assurée par un modem/routeur ASUS. À savoir, un routeur ASUS RT-AC3100 (Amérique du Nord) et sur

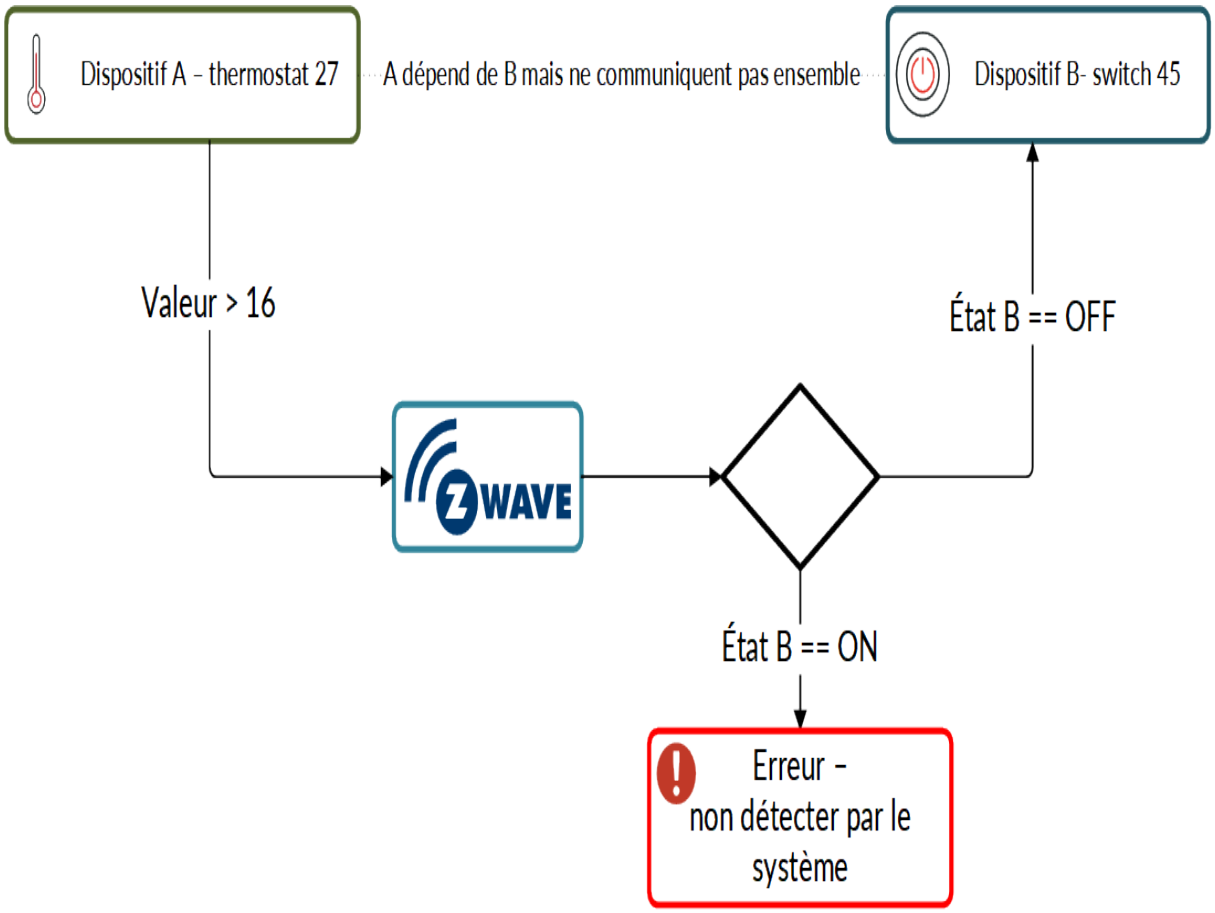


Figure 3.2 Relation entre A et B

les sites européens trois modems/routeurs ASUS (un : DSL-55U et deux DSL-52U). Le choix des appareils est motivé par les caractéristiques et le coût. En effet, ces appareils présentent trois caractéristiques principales : ils prennent tous en charge OpenVPN (serveur et client), ils permettent jusqu'à six réseaux WIFI différents avec un contrôle fin de l'accès à Internet et ils ont tous un programmateur de redémarrage.

À travers les sites, le réseau est sécurisé grâce OpenVPN. Chaque site gère un serveur OpenVPN et, à son tour, le site nord-américain AC3100, est également client des routeurs des sites européens. Ainsi, toutes les informations sensibles circulent dans les deux sens sur des canaux sécurisés.

Notons que, pour ce projet, nous nous sommes concentrer uniquement sur le site implante en Amérique du Nord. Donc tous nos traitements et analyse de données proviennent de ce dernier.

```
November 13, 2019:

[2019-11-13 22:25:45.553] [I] [core] Notification: device-info (device-temperature): "dev":"Aeotec Temperature ( #37)","l":"22.6 °C","location":7
[2019-11-13 22:25:45.560] [I] [core] ---  ZWayVDev_zway_45-0-37 performCommand processing: "0":"off"

...
2019-11-13 22:30:00.555] [I] [core] ---  ThermostatDevice_27 performCommand processing: "0":"exact","1":"level":"16"
[2019-11-13 22:30:00.564] [I] [core] Notification: device-info (device-status): "dev":" vThermostatBPP","l":"16 °C","location":7
[2019-11-13 22:30:00.734] [I] [core] ---   ThermostatDevice_27 performCommand processing: "0":"exact","1":"level":"16"

November 18, 2019:
[2019-11-18 22:23:46.423] [I] [core] Notification: device-info (device-temperature): "dev":"Aeotec Temperature ( #37)","l":"20.8 °C","location":7
...
[2019-11-18 22:30:01.657] [I] [core] ---   ThermostatDevice_27 performCommand processing: "0":"exact","1":"level":"16"
[2019-11-18 22:30:01.666] [D] [zway] RECEIVED ACK
[2019-11-18 22:30:01.667] [D] [zway] RECEIVED: ( 01 04 01 13 01 E8 )
[2019-11-18 22:30:01.667] [D] [zway] SENT ACK
[2019-11-18 22:30:01.667] [D] [zway] Delivered to Z-Wave  stack
[2019-11-18 22:30:01.682] [I] [core] Notification: device-info (device-status): "dev":" vThermostatBPP","l":"16 °C","location":7
...

November 19, 2019:
[2019-11-19 22:30:00.243] [I] [core] ---  ThermostatDevice_27 performCommand processing: "0":"exact","1":"level":"16"
[2019-11-19 22:30:00.263] [I] [core] Notification: device-info (device-status): "dev":" vThermostatBPP","l":"16 °C","location":7
[2019-11-19 22:30:00.350] [I] [core] ---  ZWayVDev_zway_45-0-37 performCommand processing: "0":"off"
...
[2019-11-19 23:23:21.324] [I] [core] Notification: device-info (device-temperature): "dev":"Aeotec Temperature ( #37)","l":"19.8 °C","location":7
[2019-11-19 23:23:21.338] [I] [core] ---  ZWayVDev_zway_45-0-37 performCommand processing: "0":"off"

November 21, 2019:
[2019-11-21 21:22:35.712] [I] [core] Notification: device-info (device-temperature): "dev":"Aeotec Temperature ( #37)","l":"21.2 °C","location":7
...
[2019-11-21 22:30:00.816] [I] [core] ---  ThermostatDevice_27 performCommand processing: "0":"exact","1":"level":"16"
[2019-11-21 22:30:00.817] [D] [zway] RECEIVED ACK
[2019-11-21 22:30:00.821] [D] [zway] RECEIVED: ( 01 04 01 13 01 E8 )
[2019-11-21 22:30:00.822] [D] [zway] SENT ACK
[2019-11-21 22:30:00.822] [D] [zway] Delivered to Z-Wave  stack
[2019-11-21 22:30:00.826] [I] [core] Notification: device-info (device-status): "dev":" vThermostatBPP","l":"16 °C","location":7
[2019-11-21 22:30:00.858] [D] [zway] RECEIVED: ( 01 07 00 13 7B 00 00 04 94 )
[2019-11-21 22:30:00.859] [D] [zway] SENT ACK
[2019-11-21 22:30:00.859] [I] [zway] Job 0x13 (SwitchBinary Get): Delivered
[2019-11-21 22:30:00.859] [D] [zway] SendData Response with callback 0x7b received: received by recipient
[2019-11-21 22:30:00.859] [D] [zway] SETDATA devices.3.data.lastSendInternal = **********
[2019-11-21 22:30:00.859] [D] [zway] SETDATA devices.3.data.lastSend = 19110130 (0x012398f2)
[2019-11-21 22:30:00.859] [D] [zway] Job 0x13 (SwitchBinary Get): success
[2019-11-21 22:30:00.859] [I] [zway] Waiting for job reply: SwitchBinary Get
[2019-11-21 22:30:00.904] [I] [core] ---   ZWayVDev_zway_45-0-37 performCommand processing: "0":"off"
...
[2019-11-21 22:46:39.849] [I] [core] Notification: device-info (device-temperature): "dev":"Aeotec Temperature ( #37)","l":"22.1 °C","location":7
...
[2019-11-22 05:22:51.527] [I] [core] Notification: device-info (device-temperature): "dev":"Aeotec Temperature ( #37)","l":"21.1 °C","location":7
```

Figure 3.3 22 :30 Extrait du log du serveur Z-Wave les 13, 18 et 19 novembre 2019 : même appareil et même horaire, mais avec un comportement différent.

3.2 MQTT Broker

Message Queuing Telemetry Transport (MQTT) est un protocole de communication/connectivité de machine à machine [1]. Il est récemment devenu une norme Organisation For The Advancement Of Structured Information Standards (OASIS) et peut être considéré comme une sorte de "Lingua Franca" permettant aux systèmes IOT de communiquer. L'idée est d'avoir un courtier MQTT, un serveur, Mosquitto [2] dans notre cas, en publiant des messages aux abonnés. Par exemple, un capteur de température ira à publier la valeur de température masurée sur un sujet "température" ; à son tour un abonné au sujet "température" recevra le message avec la valeur publiée. Chaque fois que le MQTT est pris en charge, l'intégration est possible. Cependant, il arrive parfois qu'une seule des deux parties communicantes comprenne la *Lingua Franca*. Dans ce cas, il est nécessaire d'implémenter une passerelle : un fragment de code spécifique, permettant de traduire la communication entre deux protocoles différents. Par exemple, les prises intelligentes de Leviton Decora ne supportent pas le MQTT, mais une API a été rétroconçue, et un pont peut donc être écrit reliant Decora IOT à MQTT et donc a l'univers IOT.

3.3 InfluxDB

Avoir une base de données de séries chronologiques centralisée (ou distribuée) est un élément clé dans notre système. Pour cela nous utilisons une base de données InfluxDB sous licence du MIT, car elle prend en charge la définition des données et possède un langage d'interrogation similaire à SQL. En plus, elle est suffisamment légère pour fonctionner correctement sur une Raspberry PI 3B, tout en étant puissante une fois combiné avec des dispositifs de stockage raisonnablement rapides (dans notre cas des disques durs SSD). Par exemple, voir Fig. 3.1, un script CRON vérifie toutes les dix minutes les données météorologiques (température et humidité extérieures) et les stocke dans l'InfluxDB. De même, chaque fois qu'un capteur signale un changement, les données sont enregistrées et stockées pour une analyse plus approfondie. Ceci est absolument essentiel pour identifier les problèmes dépendant du temps.

1. `http://mqtt.org/`
2. `https://mosquitto.org/`

3.4 Grafana

Grafana, de Grafana Labs[3] est un environnement de visualisation, c'est une plate-forme de surveillance et d'analyse open source. Elle s'intègre avec plusieurs bases de données. Elle peut être utilisée pour interroger, visualiser, analyser et mettre en alerte des données de séries chronologiques. Elle s'intègre avec InfluxDB et permet d'inspecter facilement les données de la base de données Raspberry PI 3B. La visualisation de grandes quantités de données est faisable, bien qu'une machine plus performante soit préférable.

3.5 SMS Gateway

Chaque fois, qu'un évènement d'alarme critique (e.g., fuite d'eau, vol) se déclenche ou n'importe quelles données du système change de valeurs important (*e.g.,* arrêt d'un sous-systéme), une notification doit être envoyée à certains terminaux mobiles (c'est-à-dire les téléphones). Il existe plusieurs solutions possibles. L'une d'entre elles consiste à envoyer un message au téléphone, en Anglais "push a message". Cependant, cela suppose que le téléphone soit connecté à Internet. Une autre option consiste à envoyer un SMS en utilisant une passerelle SMS, éventuellement avec l'aide de certains services intermédiaires (par exemple Zappier ou IFTTT). Nous avons opté pour ClickSend, qui propose des tarifs très compétitifs. ClickSend fournit une API REST, actuellement v3, et un SDK pour différents langages (*e.g.,* Python et Perl) et peut être facilement intégré dans une variété d'applications. Par exemple, sur le site principal, un script CRON vérifie l'état des connexions régulières (tout les 10 minutes) avec les sites distants et si la connexion échoue trois fois en un message SMS d'alarme est envoyé.

3.6 Anomalie fonctionnelle

Dans un système de nature distribué, comme celui étudié dans ce projet, il y a deux types d'évènements spécifiques qui présentent un intérêt particulier que nous pouvons classifiés d'anomalie fonctionnelle : Une interaction non prévue entre des composantes et la perte de mesures, de commandes ou, plus généralement, de données.

Cette situation peut donner lieu à un dysfonctionnement du système, exemple un appareil est censé annoncer son changement d'état via le MQTT, mais lorsque l'état change, aucune annonce d'état. Le changement est observé mais aucune erreur n'est détectée, ou bien l'inverse un appareil qui devrait changer d'état mais ne reçoit jamais la commande ou la valeur de déclenchement de la commande.

3. `https://grafana.com/`

Dans un système complexe, l'interaction des dispositifs est souvent le comportement attendu [33], cependant parfois nous n'obtenons pas le comportement voulu et l'interaction est destructive [34, 35].

Nous supposons qu'il existe un autre type d'interaction entre les éléments : l'absence d'interaction entre eux. c.-à-d. l'interaction s'est effectivement produite mais le résultat n'est pas observé. En effet, il peut être difficile, voire impossible, de savoir pourquoi l'interaction attendue ne s'est pas concrétisée ou si l'interaction s'est produite pour quelle raison nous n'avons pas observé son effet.

Nous appelons cela le manque d'interactions attendues et prévues, des anomalies causales. Supposons que l'événement A ait une relation imbriqué avec B : A cause B au plus tard dans un délai donné ; une anomalie causale est définie comme l'observation de la cause A, mais soit l'absence totale de B, soit l'observation de B au-delà du délai prévu. Comme, il est facile de l'imaginer, un événement non perçu, jamais délivré, peut induire une anomalie causale. De la même manière, si, en raison d'un mauvais débit et d'une latence élevée, une commande est délivrée trop tard, l'effet peut être identique ou similaire.

Nous pensons que l'anomalie causale, bien qu'elle puisse être détectée en considérant différentes fenêtres temporelles de la même ou de différentes séries temporelles, cela serait plus facile de la détecter en utilisant les relations déjà connues entre les dispositifs qui suivent déjà un scénario implémenté .

3.7 Local outlier Factor

Local outlier Factor (LOF) est une méthode non-supervisée proposée par Breuning et al. [36] pour la détection d'anomalies locales basée sur la densité du point par rapport à la densité de ses plus proches voisins. C'est une méthode qui prend en paramètre le nombre des voisins les plus proches a considéré k et calcul le degré d'aberrance d'un point p en appliquant la formule suivante : $LOF_k(p) = \frac{\sum_{o \in N(p,n)} \frac{lrd_k(o)}{lrd_k(p)}}{k}$avec $N(p,k)$ [1] l'ensemble le des k plus proches voisins de p et $lrd(p) = \frac{k}{\sum_{o \in N(p,n)} dist_k(p,o)}$

3.8 Protocole Z-Wave

Z-Wave [4] est un protocole domotique normalisé par l'ITU et soutenu par un ~~grand~~ consortium industriel (l'alliance Z-Wave). Z-Wave intègre des éléments matériels et logiciels ; ces dispositifs comprennent des interrupteurs, des ampoules, des prises, dispositifs pour la me-

4. `https://www.z-wave.com/`

sure de la température, l'humidité, des capteurs de mouvement, des serrures de porte, des alarme, *etc.* L'une des principales caractéristiques de la conception des Z-Wave est de garantir l'interopérabilité tout en créant une technologie à haut rendement et faible consommation d'énergie. Les appareils doivent pouvoir fonctionner pendant des mois ou des années avec des piles standard du commerce.

Les dispositifs Z-Wave font parties d'un réseau maillé ; ils sont essentiellement des nœuds IOT fonctionnant avec un protocole radio spécialisé et connecté à un contrôleur, passerelle vers Internet. En effet, un système Z-Wave est organisé autour d'un dispositif spécial, appelé contrôleur primaire, agissant comme un serveur. Le contrôleur primaire est également la passerelle Internet du système ; elle permet l'accès à Internet et la communication avec des nœuds individuels en utilisant HTTP (et dans certains cas l'API RESTful). Certains sont physiques (par exemple, un interrupteur), d'autres sont virtuelles (par exemple, un contrôleur du lever et coucher du soleil). Le protocole Z-Wave cré un réseau maillé où certains nœuds (selon le matériel) sont capables d'agir comme des routeurs répétant les messages pour d'autres nœuds.

Le protocole Z-Wave comporte différentes couches : physique, contrôle d'accès au support (MAC) et application. Dans un réseau en Z-Wave : un seul contrôleur peut gérer jusqu'à 232 appareils. Cependant, il pourrait y avoir plusieurs réseaux Z-Wave différents coexistant, chaque réseau étant géré par son contrôleur principal et peuvent être relié entre eux.

Pour résumer, l'infrastructure est extensible de manière similaire au protocole TCP/IP, où un pont relie deux segments IP différents. La couche MAC de Z-Wave s'occupe du routage et de l'accès au support de transmission. Elle met en œuvre une stratégie d'évitement des collisions avec retransmission aléatoire. Cela signifie qu'avant de lancer une transmission, un nœud détecte le canal du signal et, si ce dernier est déjà occupé, il fait marche arrière et attend un moment aléatoire avant de tenter à nouveau l'accès.

Le débit de la maille (vitesse de transmission) est limitée par la consommation d'énergie, le codage du signal, la largeur de bande et les vitesses standard allant jusqu'à 100 kbps. La couche MAC met en œuvre un protocole source-destination ; un nœud (contrôleur) dispose d'une table de routage (voisinage). Une table de 232 par 232 bits, si dans la cellule x,y il y en a une, cela signifie que le nœud x est capable de communiquer avec le nœud y ; x et y sont dans leur voisinage relatif. Bien entendu, le contrôleur a accès à la table de routage et communique avec tous les nœuds (secondaire). il y a deux types de noeuds secondaires : qui exécutent des commandes ou envoient de l'information et sinon ceux jouent le rôle de routeur lors de la transmission d'informations dans le réseau.

Un noeud secondaire n'a aucune information sur la table de routage, il peut seulement ré-

pondre au nœud dont il a reçu le message et ne peut donc pas initier une communication. Le noeud de routage a une connaissance partielle de la table de routage ; il est capable de répondre au nœud de qui il a reçu le message et peut envoyer des messages non sollicités à un certain nombre de nœuds prédéfinis ; il dispose également d'un itinéraire.

Le temps que prend un message pour se rendre d'un appareil à un autre varie selon l'emplacement physique de l'appareil (par exemple, derrière un mur épais), l'humidité, les interférences radio (par exemple, un four à micro-ondes) et les diverses conditions de charge du réseau. Lorsque le contrôleur principal envoie un message à un appareil, il sélectionne une voie de communication. Z-Wave peut contenir jusqu'à quatre sauts ; il n'est donc pas nécessaire d'établir des liens directs entre serveurs et appareils puisqu'un noeud de routage peut faire l'intermédiaire. Malgré ce mécanisme, la fiabilité peut encore poser problème. En effet, la communication peut échouer et le contrôleur principal peut devoir essayer d'utiliser des routes différentes afin de transmettre un message à un dispositif spécifique. En général, lorsqu'un message est envoyé, un délai d'attente est fixé ; si l'accusé de réception n'arrive pas dans le délai imparti, le contrôleur principal tentera une autre route. Si après trois tentatives différentes, il n'y a pas de succès, le serveur considère que la communication a échoué.

Il est clair que la topologie en maille et la redondance des routes jouent un rôle clé dans la fiabilité, la robustesse et la rapidité de la communication. Imaginez une ampoule activée par un mouvement. Si un mouvement est détecté, le dispositif de détection de mouvement envoie un message au serveur, qui à son tour transmet le message aux couches logicielles de l'application. Lorsque la décision d'allumer la lumière est prise, un second message doit retourner dans le réseau et atteindre l'ampoule spécifique pour l'allumer. Le nombre de sauts, l'état de la charge du réseau et les retransmissions déterminent la vitesse et la réactivité du réseau, et donc le délai entre la détection du mouvement et la lumière qui s'allume.

3.8.1 Contrôleur Z-Wave

Un contrôleur Z-Wave est principalement un pont entre les dispositifs et une passerelle avec Internet. Il existe plusieurs modèles vendus par des sociétés telles que Samsung ou Vera. Nous avons opté pour un contrôleur RaZberry, développé et vendu par Z-Wave .me [5]. RazBerry est une carte de plugin qui est installée sur un Raspberry PI donnant accès à un environnement Linux et à des outils de développement entièrement compatibles. L'une des particularités de cette solution est la disponibilité d'applications open source permettant de personnaliser et de gérer diverses tâches. Les applications sont écrites en JavaScript sur une API compilée. Elles sont réparties en un ensemble de base, installées par défaut, qui comprend par exemple

5. `https://z-wave.me/`

la gestion des horaires et des applications complémentaires telles que la passerelle MQTT, les thermostats virtuels ou les informations astronomiques [6]. L'une des applications utiles est un switch virtuel ; celui-ci peut être configuré avec la passerelle MQTT pour contrôler, via un brocker MQTT, des appareils situés en dehors du réseau Z-Wave . Inversement, la passerelle MQTT permet de contrôler des appareils en Z-Wave depuis l'extérieur du réseau de contrôle Z-Wave sans avoir besoin de se servir d'un accès URL. Une des fonctions intéressantes de la RaZberry est la possibilité de mettre à jour le micrologiciel ; depuis l'été 2019, les dernières versions donnent accès à des informations détaillées sur le niveau des paquets, y compris le routage, le bruit et la puissance du signal.

3.9 ZoneMinder

ZoneMinder est un ensemble d'applications open source fournissant un système complet de vidéo surveillance. Il permet la capture, l'analyse et le stockage d'images provenant de caméras de sécurité. Il fonctionne avec presque toutes les WEB caméras connues et possède une application de gestion Android (*i.e.,* zmNinja). La détection d'événements peut déclencher une notification d'alerte et une identification avancée des objets d'apprentissage de la machine grâce à la reconnaissance des visages ou des plaques. La composante ZoneMinder est brièvement décrite et rapportée ici pour aider à évaluer la complexité et le besoin d'intégration de divers composants et sous-systèmes hétérogènes de IOT. Comme nous pouvons l'imaginer, les changements soudains de lumière peuvent déclencher des alertes inutiles et indésirables. Il est donc crucial d'intégrer le système de sécurité ZoneMinder dans notre système domotique. Par contre, si le système Z-Wave est mis en mode "alarme" alors une caméra va détecter une personne. cette image doit être stocke et envoyer à l'usager.

6. `http://developer.z-wave.me/uri=public#/web/apps`

CHAPITRE 4 APPROCHE

Ce chapitre détaillera l'approche utilisée ainsi que les types d'anomalies détectées.

Exploratory Data Analysis (EDA) [37] est la première étape du processus d'analyse de données. Nous avons appliqué EDA sur les séries temporelles (ST) des capteurs ainsi qu'aux informations extraites des fichiers log.

4.1 Analyse des séries temporelles

Rappelons que notre système est assez complexe en terme d'analyse. Les appareils à Z-Wave (ou appareils WiFi) ont un micrologiciel propriétaire, des fonctionnalités et des mises en œuvre qui dépendent de l'entreprise. Le contrôleur Z-Wave (ou routeur WiFi et serveurs en nuage) possède également un code propriétaire. L'accès à l'API, les fichiers logs et les paquets reniflés ne donnent qu'une vue partielle des informations des logs et de l'API ce qui n'est pas suffisant pour détecter une anomalie. Nous nous retrouvons donc dans l'incapacité d'en déterminer la cause de par les nombreuses couches logicielles et les diverses interactions de facteurs entrant en jeu.

Après analyse des fichiers logs, nous avons constaté que diverses anomalies, par exemple, l'échec de la transmission de messages et le changement d'itinéraire, se traduisaient par une augmentation de l'intervalle de temps entre les messages. À partir de cette observation, nous avons décidé d'arborer une approche comparative de ces intervalles pour la détection d'anomalies.

Néanmoins, les fichiers logs ne suffisent pas toujours à repérer un problème. Prenons, pour exemple, cette situation : vous envoyez un message pour allumer le chauffage. Il n'y a pas d'erreur dans les fichiers journaux. Le message a été délivré avec succès et un thermostat en a accusé réception. Cependant, le capteur thermique reste invariable : la température n'a pas augmenté comme il se doit. En effet, un tel comportement a été observé dans notre système à plusieurs reprises. Cela implique qu'en plus des fichiers journaux, les données temporelles des capteurs doivent également être surveillées. Pour ce faire, nous avons utilisé une méthode d'observation aberrante locale suivie de techniques de regroupements.

Toujours est-il que pour détecter une anomalie, il faut d'abord l'identifier, la caractériser, la répertorier et éventuellement, résoudre tous les problèmes triviaux associés. Par exemple, une mauvaise configuration de l'adresse IP ou du mot de passe et/ou des autorisations d'application incorrectes. À ce stade, il est raisonnable de supposer que le système distribué fonctionne

comme prévu.

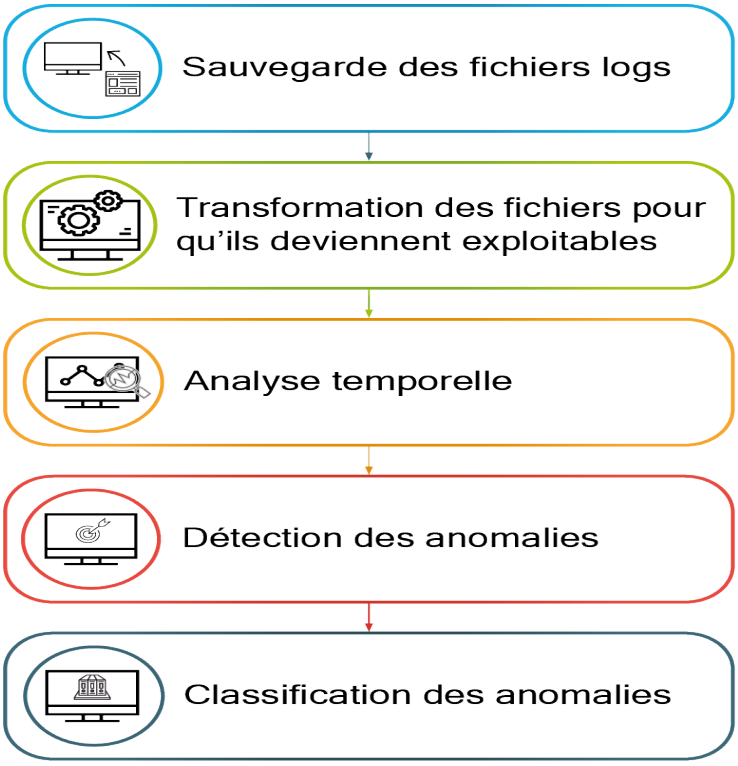


Figure 4.1 Les étapes d'analyse.

Pour détecter des anomalies temporelles, nous supposons que les étapes d'analyse suivantes sont nécessaires :

— **Transformation des fichiers logs en séries temporelles** : nous avons transformé les fichiers logs en un ensemble de séries temporelles. Ces dernières peuvent représenter des mesures enregistrées par un capteur et/ou la différence de temps entre deux événements observés. Cette démarche est importante dans la détection d'anomalies dues à la latence.

Fig 4.2 ici nous pouvons voir un exemple de transformations effectuées. La "transformation 1" représente notre base de traitement des fichiers. Nous avons décidé de diviser nos logs en jours de la semaine. Cela étant fait dans l'optique de comparer des séries temporelles d'une même journée. En effet, notre système est implanté dans un environnement réel ce qui implique une certaine routine. De ce fait, comparer un dimanche à un lundi relèverait de l'erreur et serait non-pertinent en considérant la réalité du quotidien. Ce cas est vraiment spécifique à notre environnement de travail. Ensuite, la "transformation 2" représente le cas de figure où nous nous intéressons à un

```
Octobre 26, 2019:
original:

[2019-10-26 06:26:02.660] [D] [zway] SETDATA devices.36.data.lastReceived = 0 (0x00000000)
[2019-10-26 06:26:02.660] [I] [zway] Node 36:0 CC Security: Received a secure message
[2019-10-26 06:26:02.661] [D] [zway] SETDATA devices.36.instances.0.commandClasses.152.data.firstPart =
**********
[2019-10-26 06:26:02.661] [I] [zway] Node 36:0 CC Security: passing decrypted packet to application level:
[ 31 05 05 01 3c ]
[2019-10-26 06:26:02.661] [D] [zway] SETDATA devices.36.instances.0.commandClasses.49.data.5.deviceScale =
0 (0x00000000)
[2019-10-26 06:26:02.661] [D] [zway] SETDATA devices.36.instances.0.commandClasses.49.data.5.scale =
0 (0x00000000)
[2019-10-26 06:26:02.664] [D] [zway] SETDATA devices.36.instances.0.commandClasses.49.data.5.scaleString = "%"
[2019-10-26 06:26:02.664] [D] [zway] SETDATA devices.36.instances.0.commandClasses.49.data.5.val =
60.000000
[2019-10-26 06:26:02.664] [D] [zway] SETDATA devices.36.instances.0.commandClasses.49.data.5 = Empty
[2019-10-26 06:26:03.275] [I] [core] Notification: device-info (device-temperature):
{"dev":"Temperatura Soggiorno (#36)","l":"18.1 ˚C","location":2}
. . .


Transformation 1 :

2019-10-26;06;26;02.660;6;1572085562.66;36;SETDATA devices.36.data.lastReceived = 0 (0x00000000)
2019-10-26;06;26;02.660;6;1572085562.66;36;Node 36:0 CC Security: Received a secure message
2019-10-26;06;26;02.661;6;1572085562.661;36;SETDATA devices.36.instances.0.commandClasses.152.data.firstPart =
**********
2019-10-26;06;26;02.661;6;1572085562.661;36;Node 36:0 CC Security: passing decrypted packet to application level:
[ 31 05 05 01 3c ]
2019-10-26;06;26;02.661;6;1572085562.661;36;SETDATA devices.36.instances.0.commandClasses.49.data.5.deviceScale =
0 (0x00000000)
2019-10-26;06;26;02.661;6;1572085562.661;36;SETDATA devices.36.instances.0.commandClasses.49.data.5.scale =
0 (0x00000000)
2019-10-26;06;26;02.664;6;1572085562.664;36;SETDATA devices.36.instances.0.commandClasses.49.data.5.scaleString =
__ % __
2019-10-26;06;26;02.664;6;1572085562.664;36;SETDATA devices.36.instances.0.commandClasses.49.data.5.val =
60.000000
2019-10-26;06;26;02.664;6;1572085562.664;36;SETDATA devices.36.instances.0.commandClasses.49.data.5 = Empty
2019-10-26;06;26;03.275;6;1572085563.275;temperature;Notification: device-info (device-temperature):
{ __ dev __ : __ Temperatura Soggiorno (#36)__  __comma__  __ l __ : __ 18.1  C __  __comma__  __ location __ :2}
. . .

Transformation 2 :

2019-10-26,06,06,02.394,6,1572084362.394, Temperatura Soggiorno,36,2,18
2019-10-26,06,16,04.693,6,1572084964.693, Temperatura Soggiorno,36,2,18.1
2019-10-26,06,21,02.189,6,1572085262.189, Temperatura Soggiorno,36,2,18
2019-10-26,06,26,03.275,6,1572085563.275, Temperatura Soggiorno,36,2,18.1
2019-10-26,06,36,02.788,6,1572086162.788, Temperatura Soggiorno,36,2,18
2019-10-26,06,46,04.314,6,1572086764.314, Temperatura Soggiorno,36,2,18.1
2019-10-26,06,56,04.484,6,1572087364.484, Temperatura Soggiorno,36,2,18
2019-10-26,07,01,02.954,6,1572087662.954, Temperatura Soggiorno,36,2,18.1
2019-10-26,07,21,01.338,6,1572088861.338, Temperatura Soggiorno,36,2,18
2019-10-26,07,31,01.240,6,1572089461.24, Temperatura Soggiorno,36,2,18.1
2019-10-26,07,51,01.703,6,1572090661.703, Temperatura Soggiorno,36,2,18
2019-10-26,07,56,01.132,6,1572090961.132, Temperatura Soggiorno,36,2,18.1
2019-10-26,08,21,01.531,6,1572092461.531, Temperatura Soggiorno,36,2,18
2019-10-26,08,31,00.850,6,1572093060.85, Temperatura Soggiorno,36,2,17.9
. . .
```

Figure 4.2 transformation des logs

capteur en particulier. Cet exemple nous montre quelques valeurs enregistrées le long de la journée du 26 octobre pour le capteur #36 "Temperatura Soggiorno" qui se situe à la localité 2. Pour nos analyses, nous utilisons la localité (pièce) et les caractéristiques de chaque capteur au lieu d'une recherche par ID puisque, malheureusement, cette valeur peut varier au cours du temps.

— **Extraction des données des dispositifs** : l'objectif est d'obtenir des données d'informations ou des valeurs mesurées des dispositifs marqués dans le temps. Par exemple, nous pouvons être intéressés par l'enregistrement et le stockage des valeurs de température et d'humidité d'un capteur spécifique. Nous pouvons également nous intéresser au temps d'activation d'un appareil. Une augmentation du temps d'activation peut représenter une anomalie tout comme son contraire.

— **Détection d'anomalies discrète ou continue dans un intervalle** : où l'objectif est de trouver à l'intérieur d'une même série temporelle des valeurs inattendues ou aberrantes des sous-séquences. Par exemple, en raison d'une erreur de réseau, un thermomètre peut signaler une température erronée. Notez que si deux ou plusieurs valeurs aberrantes sont signalées en séquence et dans un court laps de temps, il s'agit d'une anomalie continue dans un intervalle.

— **Détection d'anomalies basée sur la connaissance** : nous cherchons ici à identifier les anomalies fonctionnelles au moyen d'un ensemble de règles et d'heuristiques. Par exemple, si à un moment donné de la journée, le chauffage doit être coupé, à partir de ce moment, la température doit diminuer (en supposant que la température extérieure est plus basse).

— **Détection d'anomalies fantômes** : c'est une anomalie non visible dans les logs mais qui reste un phénomène non expliqué et requière plusieurs tests et observations pour être détectés. Il en existe deux types : ceux qui sont liés à des anomalies causales et ceux qui ne le sont pas. Prenons par exemple un thermostat qui indique la température, avec une précision de 1 degré. Supposons en outre que le thermostat doit stocker les valeurs mesurées dans la base de données. Si dans la base de données nous trouvons les valeurs de 20 degrés Celsius puis de 24 degrés Celsius, nous pouvons en déduire que quelque chose s'est mal passé car certaines valeurs intermédiaires auraient dû être enregistrées.

— **Détection d'anomalies dans des séries temporelles croisées** : Nous cherchons ici à comparer et à contraster plusieurs séries temporelles afin de repérer les comportements inattendus. Cela peut nécessiter des outils assez complexes, mais il existe des cas spécifiques où les anomalies sont facilement détectées. Revenons à l'exemple de la lumière activée par un mouvement qui ne devrait fonctionner qu'à une certaine heure

de la nuit. Nous pouvons considérer cela comme un exemple de problème d'anomalie causal.

4.2 Méthode de détection

Au moment où nous écrivons ces lignes, nous avons mis en œuvre et testé dans notre système de mise en place une version préliminaire, une preuve de concept et diverses analyses qui ont permis de réaliser les étapes décrites. Dans cette partie, nous allons décrire plus en détail les outils et les analyses que nous avons utilisés.

4.2.1 Analyse des fenêtres de longueur fixe

Nous avons choisi d'utiliser cette approche suite à une observation des logs qui a montré que des fois nous avons une densité de message plus importante dans certaines journées et horaires. Nous avons noté des différences de dix fois plus que la normal (observer habillement). Pour cela nous avons émis l'hypothèse que nous pouvons avoir une anomalie vu que normalement pour des journées similaires avec un comportement assez routinier nous devrons pas avoir un pic d'activités si importantes surtout à des horaires peu probables comme a minuit.
Comme au début de tout test statistique, nous établissons d'abord une hypothèse nulle et une hypothèse alternative. Notre hypothèse nulle est qu' **il n'y avait pas d'anomalies dans notre système**, alors que notre hypothèse alternative est que certaines anomalies sont présentes.

En théorie, nous devrions disposer d'un ensemble de fenêtres que nous considérons comme normales. Pour éviter d'étiqueter ce qui est normal ou anormal, nous nous appuyons sur le concept de valeur aberrante. Si une fenêtre de distribution de fréquence est un cas aberrant entre la distribution des fenêtres donnée, nous considérons que ces fenêtres spécifiques correspondent à une anomalie. En pratique nous comparons chaque fenêtre dans un ensemble donné, à toutes les autres : si aucune fenêtre n'est similaire, nous considérons cette fenêtre comme anomalie. Un tel critère peut être trop restrictif et si nécessaire nous pourrions utiliser la distribution des fenêtres similaires (pour une fenêtre donnée combien de similarité nous avons trouvé), jumelées à la définition de valeurs aberrantes pour cette même distribution.

Nous avons donc utilisé trois tests (likelihood ratio, Wilcoxon et Kolmogorov-Smirnov) pour repérer tout changement, quelle qu'en soit la direction (pics d'activités ou absence d'activités). Nous suivons les étapes suivantes Fig 4.3 :

— **Sélection des fenêtres** : les logs traités sont regroupés dans des fenêtres de taille égale. Pour exemple une fenêtre est générée chaque 1000 échantillons c.-à-d 1000 lignes

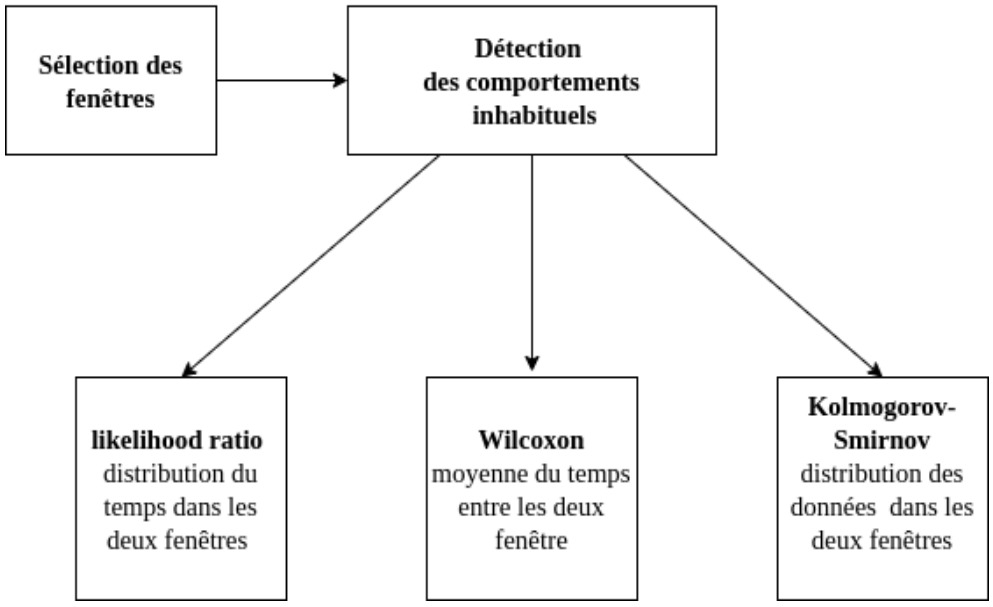


Figure 4.3 Les étapes d'analyse des fenêtres de longueur fixe

du fichier log. Par la suite, chaque fenêtre est utilisée pour extraire les données à comparer (pour exemple, temps entre les événements c.-à-d la différence du temps entre deux lignes consécutives)

— **Détection des comportements inhabituels** :les fenêtres sont ensuite comparées deux à deux (c.-à-d chaque fenêtre versus chaque autre fenêtre) en utilisant trois méthodes statistiques. Nous utilisons le test likelihood ratio, le test de Wilcoxon et le test de Kolmogorov-Smirnov.

Pour que notre analyse soit plus précise nous avons découpé nos logs en journée de la semaine et en heure puis nous avons pris des intervalles de taille fixe avec la différence de temps (ΔT) enregistrer entre chaque message. Etant donné que, nous ne pouvons pas avoir un oracle pour ce genre de comparaison, ça reste un système en temps réel et le modèle parfait avec le comportement attendu n'est pas modélisable. C'est pour cela que nous avons opté d'utiliser ces trois méthodes pour la prise de décision :

The likelihood ratio test [1] Chacune des deux fenêtres comparées est divisée en bacs et la distribution des fréquences est déterminée. Ensuite, nous testons si la fréquence des événements dans les deux fenêtres est différente. Si aucune autre fenêtre n'a la même distribution d'histogramme, nous pouvons en déduire que cette fenêtre est une exception.

1. `https://www.linkedin.com/pulse/how-detect-time-series-anomalies-when-stable-pattern-alain-fourmigue/`

Wilcoxon test Ce test non-paramétrique est basé sur le calcul de la somme des rangs pour les différences négatives et positives entre les échantillons. Le test permet d'identifier la direction et le degré de différence entre les groupes d'échantillons et de savoir si la médiane est plus décalée dans une direction par rapport à l'autre.

Kolmogorov-Smirnov statistic on 2 samples Ce test est basé sur le calcul des paramètres D-statistc et permet de déterminer si deux échantillons de données sont issus de la même distribution, c'est-à-dire s'il existe des différences statistiquement significatives entre eux. Rappelons que les fenêtres sont constituées par les différences de temps entre deux événements successifs ; nous vérifions donc que les deux fenêtres sont générées par la même distribution statistique.

Pour chacune de ces méthodes, nous avons développé des scripts python, qui acceptent des données groupées à l'entrée et renvoient la valeur p pour le test. Nous la comparons avec notre seuil (alpha = 0,05) pour rejeter ou non l'hypothèse nulle (absence d'anomalies). C.-à-d la distribution des données dans les deux fenêtres est la même (*i.e.,* si ce n'est pas le cas, nous la rejetons et nous avons une anomalie).

Après avoir obtenu les résultats des trois méthodes (trois "experts") : Chaque méthode est considérée comme un expert et nous procédons à un scrutin majoritaire pour la classification (présence ou non d'une anomalie) comme le montre la figure 4.4.

4.2.2 Détection des valeurs aberrantes basée sur la connaissance et l'heuristique

Dans de nombreuses situations, les données mesurées sont liées soit à l'environnement, soit aux lois physiques, soit aux caractéristiques du capteur. Par exemple, il est difficile d'imaginer une température inférieure à 0 Kelvin, ou une une humidité supérieure à 100 %. Ces contraintes physiques définissent les limites par rapport auxquelles les mesures du capteur doivent être vérifiées. De plus, les applications peuvent avoir des contraintes entre les événements. Prenons par exemple le chauffage d'un bâtiment ; en hiver, lorsque le chauffage est éteint, la température à l'intérieur du bâtiment devrait diminuer. Nous pouvons connaître avec précision l'heure à laquelle le chauffage est éteint (allumé) et vérifier le comportement attendu. Considérons la figure 4.5, elle indique le temps en secondes entre la détection de mouvement et l'allumage de la lumière. L'histogramme a été tronqué, cependant, nous observons une distribution bi-modale avec des valeurs élevées et assez étalées. Les deux dispositifs (capteur de détection de mouvement et lumière) se trouvent à l'intérieur du même réseau Z-Wave, situé à une dizaine de mètres au maximum du contrôleur. Cette connaissance et la distribution des données sont utiles pour définir des heuristiques afin de mettre en évidence les anomalies.

```
#LOGStart,parsed/2019-04-17_3/H00-log-1.csv.dif, 04, 17, 3, 00
#LOG,parsed/2019-04-17_3/H00-log-1.csv.dif, parsed/2019-04-24_3/H00-log-1.csv.dif, 04, 17, 3, 00, 04, 24, 3, 00
0
1
1
1
0
0
0
1
0
1
1
1
1
1
0
0
#Summary,parsed/2019-04-17_3/H00-log-1.csv.dif,parsed/2019-04-24_3/H00-log-1.csv.dif,1000,9,7
...
```

Figure 4.4 exemple de résultat des trois experts : 1 correspond a une anomalie

Afin de détecter automatiquement les anomalies, nous avons utilisé la méthode des facteurs aberrants locaux, qui consiste à identifier les régions de données de densité similaire. Ici, la densité est caractérisée par la proximité de la valeur donnée par rapport aux valeurs voisines. Les régions de plus faible densité sont appelées aberrantes et classées comme des anomalies. Nous avons observé des anomalies discrètes (une valeur aberrante dans une région) et continues (plusieurs valeurs aberrantes dans une région).

Les anomalies, sur plusieurs séries temporelles, sont détectées par un ensemble d'heuristiques et de règles encoder sur les informations des dispositifs (*e.g.,* l'humidité, le capteur de luminosité rapporte une lecture toutes les cinq minutes), les phénomènes contrôlés (*e.g.,* le réservoir de chauffage de l'eau, où nous contrôlons l'énergie mesurée par l'interrupteur) et les effets (*e.g.,* si le switch du chauffage est "Off" la température devrait diminuer).

4.2.3 Analyse exploratoire des données via InfluxDB et Grafana

Nous sommes surtout intéressés à comprendre le schéma général de ces séries temporelles univariées et multivariées. Par exemple, un dispositif multicapteur peut capturer température, humidité, vibrations, luminosité et ultraviolets. Nous pouvons nous concentrer sur une seule des séries temporelles, *e.g.,* humidité, et étudier la tendance ou la présence d'anomalies. Dans une application IOTdistribuée avec de nombreux CPS, il y a un flux continu de données. Comme nous l'avons déjà dit, InfluxDB fonctionne sur un matériel assez peu coûteux (*e.g.,* Raspberry PI) et peut être facilement couplé avec Grafana. Grafana permet de choisir des fenêtres de temps (*e.g.,* dernières 12 heures ou jours ou début et fin de fenêtres de temps).

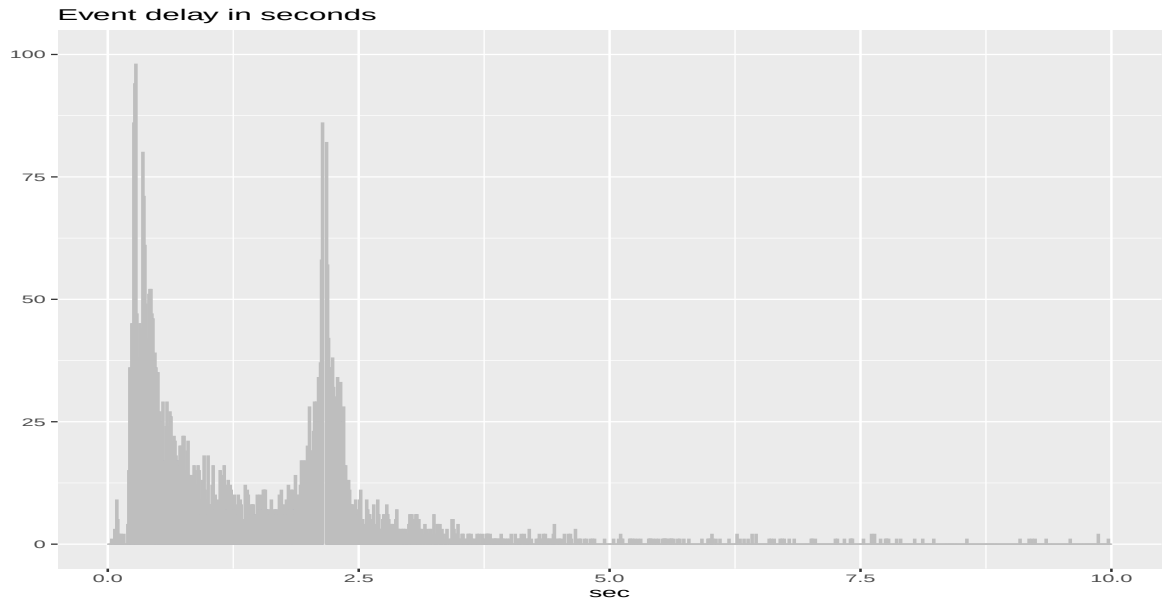


Figure 4.5 Le délai entre la détection d'un mouvement et l'allumage de la lumière (statistique sur cinq mois).

Plusieurs séries temporelles peuvent être superposées sur un graphique.

Par exemple, la Fig 4.6 montre clairement un cas extrême, probablement en raison d'une erreur de transmission. La visualisation peut être couplée à l'analyse des valeurs aberrantes et à l'analyse des tendances. Par exemple, lorsque le chauffage est éteint, il est raisonnable de supposer la baisse de température. Pour mettre en œuvre les analyses de tendance, nous appliquons les statistiques de tendance des séries temporelles disponibles dans le package `trend`. En particulier, la pente de Sen pour le taux de variation linéaire [38], la statistique de Lanzante [39] et le test de tendance de Cox et Stuart [40]. Nous avons trouvé la pente de Sen particulièrement utile et intuitive, puisque le test renvoie la pente estimée, y compris les diagnostics (c'est-à-dire la valeur p et l'intervalle de confiance).

4.3 Solution aux anomalies

Une fois qu'une anomalie est détectée, l'étape suivante consiste à la corriger. Nous devons faire une distinction entre deux cas distincts : (1) la situation où les actions d'un dispositif ont des effets mesurables sur le monde réel (en supposant que nous ayons également un ou plusieurs capteurs de surveillance de ces effets) ; et (2) les dispositifs qui effectuent certaines actions de manière autonome et pour lesquels un capteur de surveillance n'existe pas ou n'est pas pertinent (*e.g.,* une détection de fumée ou une alarme de fuite d'eau).

Figure 4.6 Grafana EDA : un extrait de la série temporelle des températures

Dans ce dernier cas, les pertes de données sont critiques, difficiles ou impossibles à corriger. Ici, le traitement est primordial. Notre stratégie est basée sur l'amélioration de la topologie de maillage, du RSSI et de la qualité générale des routes mais aussi nous pouvons doubler le nombre de dispositifs qui sont vraiment critique *e.g.,* détecteur de fumée . Ces nœuds critiques doivent être configurés pour communiquer sur les chemins de maillage les plus fiables et les plus robustes. Un contrôleur Z-Wave tente à trois reprises de fournir le message sur un maximum de sept itinéraires différents avant de marquer le message comme ayant échoué. Avoir différent chemins robustes minimisent le risque de pertes de données ayant des conséquences catastrophiques.

Pour les autres dispositifs, nous supposons une stratégie inspirée de l'analyse bien connue du fichier journal `fail2ban`[2]. `fail2ban` est un outil de détection d'intrusion Unix qui analyse les fichiers journaux de diverses applications et utilise les connaissances encodées dans un ensemble d'heuristiques pour se protéger contre les attaques. Dans notre contexte, une anomalie est comme une attaque. Il suffit d'analyser les journaux, de renifler les données et d'utiliser les connaissances disponibles (*e.g.,* calendrier prévu) pour coder, par catégorie et rôle de dispositif, des règles heuristiques pour détecter les anomalies et effectuer des actions correctives. L'action doit prendre en compte le type d'anomalie, le dispositif, les temps de réaction significatifs et l'état du trafic. Par exemple, pour un capteur de température, qui

2. `https://www.fail2ban.org`

signale sa valeur toutes les cinq minutes, une seule donnée erronée peut ne nécessiter aucune action. Si un appareil signale une fuite, il faut immédiatement effectuer une deuxième lecture de l'appareil pour confirmer avec un temps de réaction très court. En substance, l'action et le temps de réaction dépendent de l'importance du dispositif (c'est-à-dire s'il s'agit d'un dispositif critique) et de l'urgence du problème.

CHAPITRE 5 ÉTUDE DE CAS

Dans ce chapitre nous présenterons notre système de test pour tous les travaux effectués. La figure 3.1 donne un portait de haut niveau du système. Tout au long des paragraphes suivants, nous détaillerons les aspects saillants du système domotique ; la collecte de données, leurs utilisations et les processing opérés.

5.1 Installation et configuration du système Z-Wave

Un système d'automatisation Z-Wave est composé de deux principaux volets qui ne sont autres que le matériel "Rasberry Pi" et le logiciel "Z-Wave.me". La méthode la plus simple pour le faire fonctionner serait d'aller dans un premier temps sur le site web [1], télécharger l'image prête à être gravée et/ou exécutée, flasher une carte et redémarrer votre pi. Ensuite, dans un second temps, commencer à installer les composants (logiciels et matériels).

Une alternative serait d'installer un distribution Linux Raspbian à partir de zéro et d'ajouter zwave.me par-dessus. Dans les deux cas, la démarche est effective. Cependant, il faudra souligner que la deuxième option demande davantage de compétences techniques. Cela la rend alors inaccessible, voire inappropriée, pour tout usager amateur, n'ayant pas de connaissances informatiques préalables et/ou une familiarité avec le système d'exploitation Linux.

5.1.1 Composants essentiels

Pour reproduire le même système-test que celui utilisé lors de ce projet, il nous faudra :

- Un clavier, une souris et un écran HDMI.
- Une Raspberry pi 3B (ou idéalement une 4B). La taille de la RAM est négligeable, mais plus elle sera importante, mieux ce sera. Il faudra noter que la Rasberry Pi4 a un micro HDMI et une alimentation via USB-C ; le modèle précédent utilisait, quant à lui, un port micro USB.
- Une carte microSD serait préférable à partir de la taille de 32 Go.
- Un contrôleur Z-Wave.me ; vous pouvez, dans ce cas, acheter une carte RaZberry [2].
- Un disque SSD n'est pas vraiment obligatoire mais serait un plus ; un disque SATA 100 Go ferait bien l'affaire avec un adaptateur SATA vers USB. À noter que les différents modèles de disques SSD peuvent demander plus ou moins de courant. Malheureuse-

1. `https://z-wave.me/z-way/download-z-way/`
2. `https://z-wave.me/products/razberry/`

ment, la Raspberry pi est très sensible à cet aspect : le courant maximal disponible pour un port USB est de 1A. Il faudra, donc prévoir, si nécessaire, un adaptateur avec alimentation ou une barre USB alimentée. Nous utilisons, dans notre système, un disque SSD Samsung T3 de 250 Go qui est faible en consommation énergétique. Nous ajouterons, qu'après observation, nous avons remarqué que les disques SSD Crucial de 1 Tetraoctets requièrent une source d'énergie supplémentaire.

— S'il s'agit d'un système à domicile, il faudra avoir un routeur qui supporte le VPN, nous vous suggérons fortement de choisir un routeur avec un serveur OpenVPN pré-compilé.

Après avoir acquis tout le matériel nécessaire, vient l'installation du système d'exploitation. En général, les nouvelles cartes arrivent déjà prêtes à l'emploi. Si ce n'est pas le cas, l'installation est assez simple. Il suffira de télécharger et d'installer la dernière distribution de Raspbian [3] de charger, par la suite, l'image sur une carte microSD et enfin, de démarrer. Après avoir fini la configuration initiale (le fuseau horaire, la langue, le clavier, la connexion WIFI, le mot de passe), il faudra activer le mode SSH qui nous permet d'accéder à distance à la Rasperry. Pour cela, il est suffisant d'ouvrir un "shell" et lancer la commande `sudo raspi-config`. Dans la section "Interfaces", choisir "ssh" et choisir d'activer. Il faut aussi se rappeler que l'usager standard pour ces systèmes est "pi", donc, pour des raisons de sécurité, il est envisageable de modifier le mot de passe par défaut. Dans tous les cas, il faudra suivre les étapes suivantes : 1) Ajouter un nouvel usager, 2) ajouter cet usager au groupe `sudo` ; et 3) Désactiver l'usager "pi". Une fois terminé, dans le but créer un environnement plus convivial, nous pourrons ajouter les paquets suivants :

— InfluxDB
— Grafana
— Mosquitto et Mosquitto-clients : serveur et client MQTT
— Python
— Serveur SSMTP ; ancien serveur de messagerie qui peut s'avérer très pratique pour envoyer du courriel.
— screen : offre la possibilité de lancer et d'utiliser plusieurs sessions shell à partir d'une seule session ssh ; il est possible de détacher (et rattacher) la session et de laisser la computation s'exécuter aussi longtemps que nécessaire.

À noter que les composantes ci-dessus ne sont pas forcements nécessaires, pour le contrôle d'une maison, cependant, elles sont pratiques pour exécuter des tâches répétitives, de longue durée, stoker les données et créer des ponts vers les dispositifs WIFI.

3. `https://www.raspberrypi.org/downloads/raspberry-pi-desktop/`

5.1.2 Interface graphique de Z-Wave

Avec l'interface graphique de Z-Wave, nous pouvons avoir accès à plusieurs fonctionnalités indépendamment de ce que nous voulons faire. L'interface a deux modes : l'un est le mode utilisateur et l'autre, le mode expert.

Mode Utilisateur

En mode utilisateur, nous pouvons créer des pièces et leur affecter des dispositifs afin d'avoir une copie de notre espace en virtuel comme le montre la figure 5.1. Notons que les dispositifs,

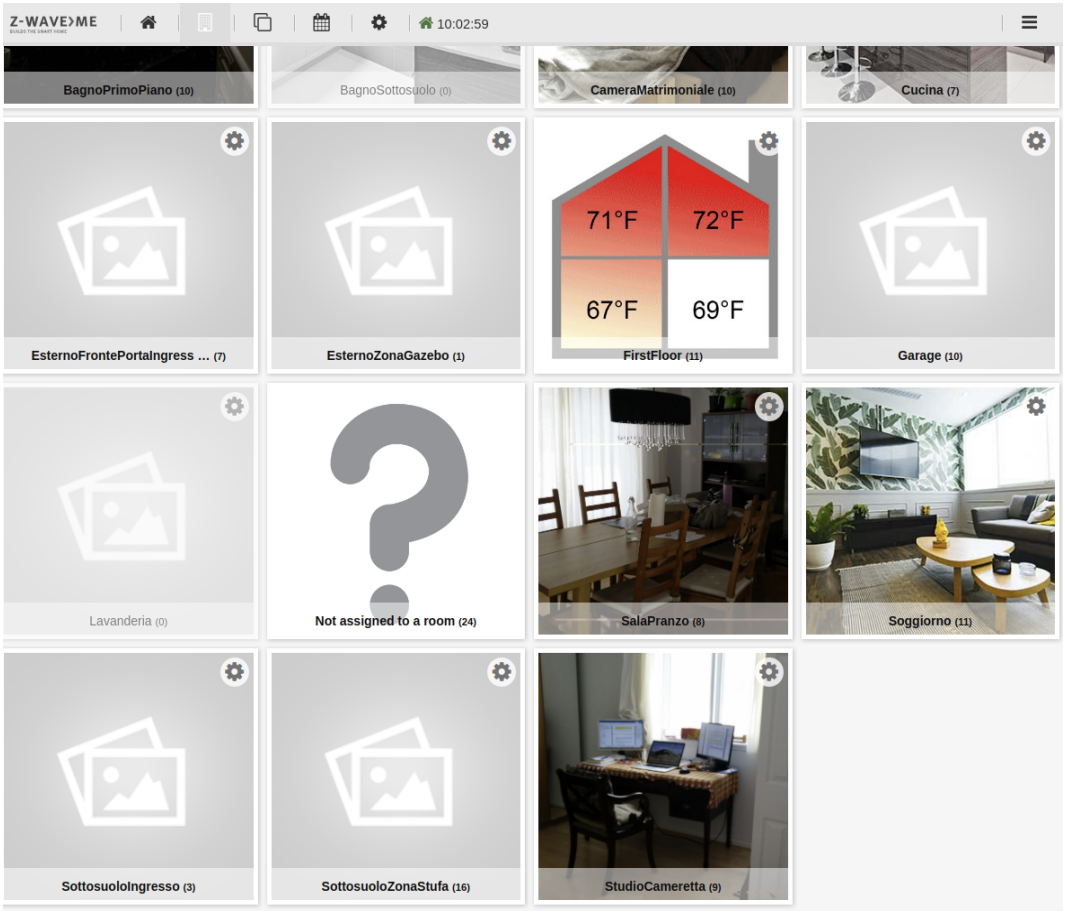


Figure 5.1 vue d'ensemble des pièces

figure 5.2, sont ajoutés un par un et un nom par défaut est assigné. L'usager peut en tout cas changer le nom automatiquement attribué. Chaque dispositif ne peut être lié qu'à une seule pièce (ou ajouter à l'écran "maison" - landing homepage).

Ce qui est aussi intéressant, ce sont les règles personnalisées que nous pouvons affecter grâce à l'option automatisation de l'interface comme nous le montre ci-dessous la figure 5.3. Pre-

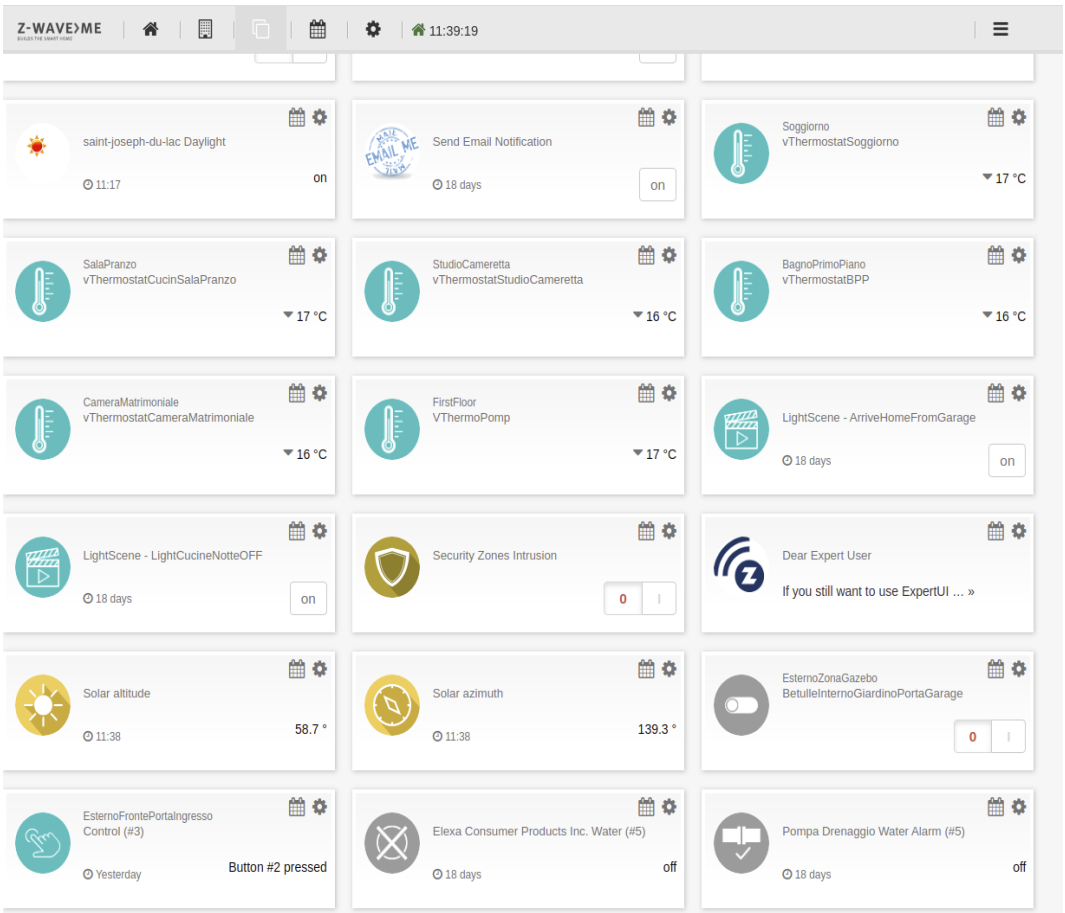


Figure 5.2 listes des dispositifs

nons, pour exemple, le contrôle du chauffage. Nous pouvons choisir l'heure d'ouverture et de fermeture, ainsi que la température minimale à ne pas dépasser. Nous pouvons, également, créer des scènes qui nous permettent d'allumer la lumière, uniquement à la tombée du jour, lors de la détection d'un mouvement. Il nous suffit de relier chaque capteur de mouvement à une lampe intelligente, comme nous pouvons relier les capteurs de fumée à une alarme sonore et a un message instantané envoyé sur notre téléphone. Plusieurs scénarios sont envisageables, c'est à l'utilisateur d'adapter son environnement selon ses besoins.

Ces actions sont formulées sous cette forme logique :

$$IF\ (évènement)\ AND/OR/NOT\ (condition)\ THEN\ (action).$$

Ou bien grâce à des applications de contrôle comme le montre la figure 5.4 cet exemple nous montre l'activation de la thermopompe à chaque jour a 5 :25 du matin.

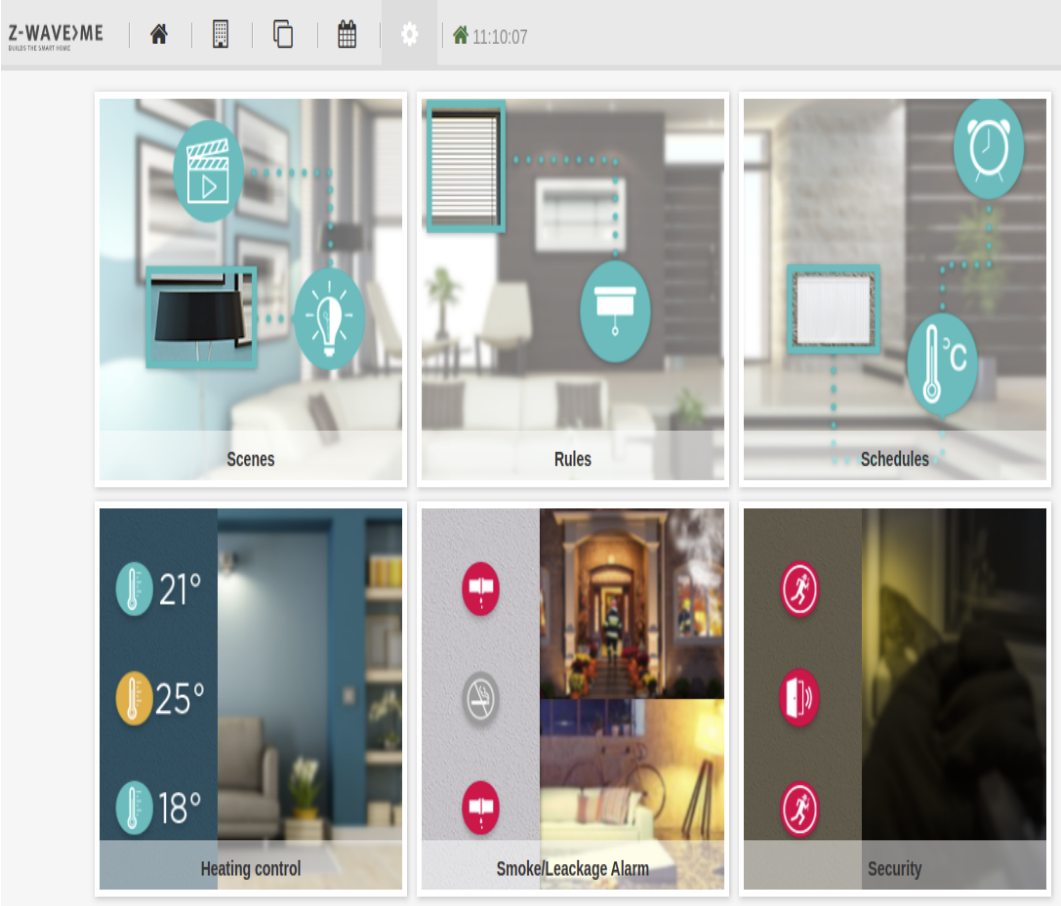


Figure 5.3 automatisation des pièces

Mode Utilisateur expert

En mode expert, nous avons plus de flexibilité mais l'interface est plus compliquée à manier. Nous pouvons inclure et/ou exclure des dispositifs, leurs états et les liaisons qu'ils ont entre eux.

Les dernières versions du micrologiciel Z-Wave donnent accès à des informations et des analyses avancées sur les réseaux. Avoir accès aux détails du trafic Z-Wave ou sniffer le trafic du réseaux Z-Wave n'est pas une mince affaire et les nombreux éléments que nous avons testés ont, presque tous, échoué, lamentablement. Alors, mieux vaut utiliser le logiciel d'analyse du serveur.
Le sniffer sur la fig 5.5 nous permet d'avoir une idée plus détaillée sur le trafic du réseaux *e.g.,* la sources, la destination, les noeuds intermédiaire.

Notez bien : la fiabilité de ce sniffing n'est pas garantie. Nous avons pu constater des observations incohérentes que nous ne pouvons pas expliquer à ce stade. Exemple si nous prenons

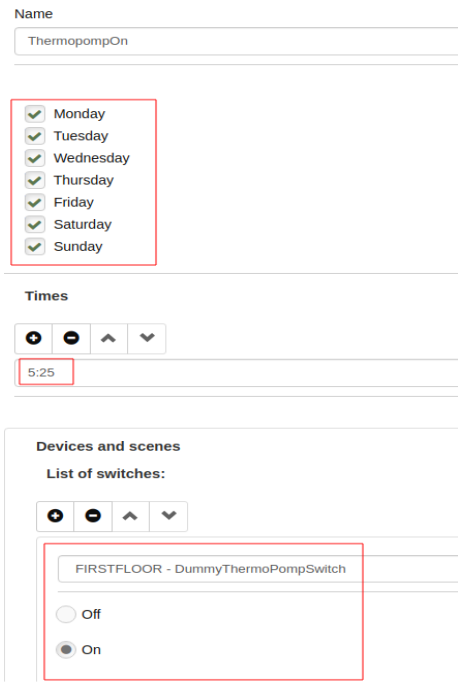


Figure 5.4 réglage de la thermopompe

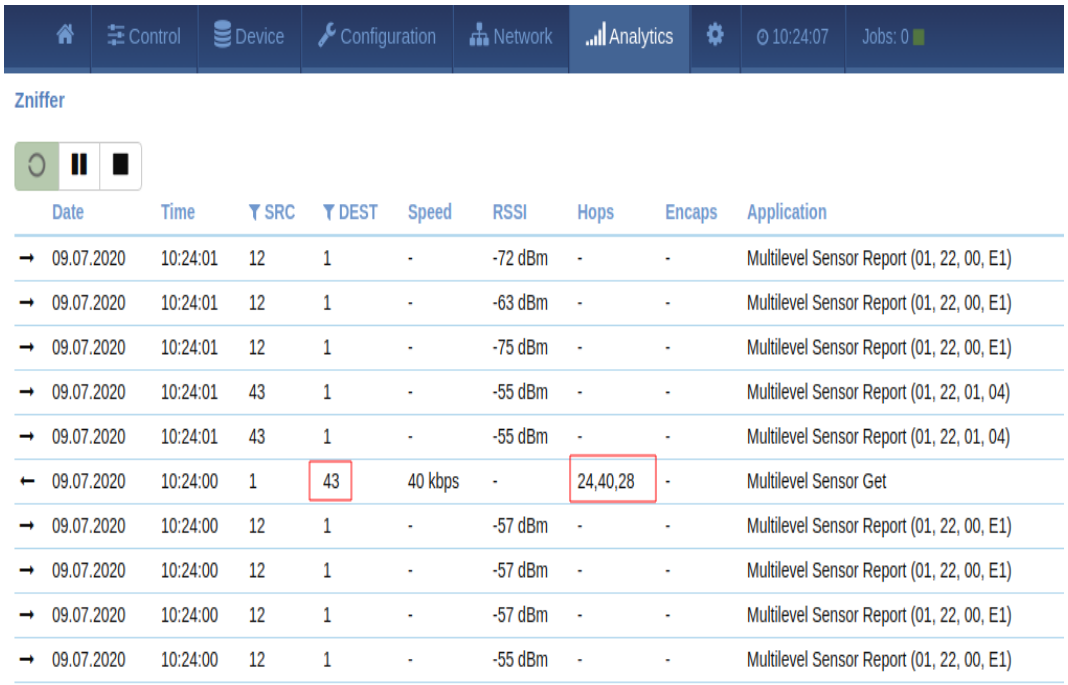
le tableau 5.5, le message envoyé par le serveur a 10 :24(9 juillet) au noeud 43 est passé par les noeuds 24,40,28 sauf que la fig 5.6 de la même journée, nous montre que ce chemin est impossible. la fig 5.6 et la fig 5.7 prouvent que le réseau change constamment.

L'interface expert nous permet, aussi, de vérifier le statut de chaque dispositif comme le montre la figure 5.8. Nous pouvons vérifier le niveau de la batterie du capteur en faisant un test de connectivité. Nous pourrons, ainsi, voir si le dispositif répond comme il le faut et vérifier les dates des derniers tests du serveur vu qu'il y a des mises à jours automatiques.

5.2 Préparation des données

Notre collecte de données a été effectuée sur une période approximative de 13 mois. Plus précisément, du 31 mars 2019 au 16 mai 2020. Cette procédure a été réalisée grâce à une sauvegarde journalière du fichier log Z-Wave. Notons qu'il faut prêter attention à l'heure de la sauvegarde pour ne pas se retrouver avec deux bouts de journées différentes. Il faut, également, toujours vérifier les dates, lors de la transformation des fichiers.log en fichiers.csv pour ne pas avoir d'incohérences.

Plusieurs informations sont automatiquement conservées par l'environnent que on a mis en place. Premièrement, chaque jour à minuit, l'image du système au complet est enregistrée.



	Date	Time	SRC	DEST	Speed	RSSI	Hops	Encaps	Application
→	09.07.2020	10:24:01	12	1	-	-72 dBm	-	-	Multilevel Sensor Report (01, 22, 00, E1)
→	09.07.2020	10:24:01	12	1	-	-63 dBm	-	-	Multilevel Sensor Report (01, 22, 00, E1)
→	09.07.2020	10:24:01	12	1	-	-75 dBm	-	-	Multilevel Sensor Report (01, 22, 00, E1)
→	09.07.2020	10:24:01	43	1	-	-55 dBm	-	-	Multilevel Sensor Report (01, 22, 01, 04)
→	09.07.2020	10:24:01	43	1	-	-55 dBm	-	-	Multilevel Sensor Report (01, 22, 01, 04)
←	09.07.2020	10:24:00	1	43	40 kbps	-	24,40,28	-	Multilevel Sensor Get
→	09.07.2020	10:24:00	12	1	-	-57 dBm	-	-	Multilevel Sensor Report (01, 22, 00, E1)
→	09.07.2020	10:24:00	12	1	-	-57 dBm	-	-	Multilevel Sensor Report (01, 22, 00, E1)
→	09.07.2020	10:24:00	12	1	-	-57 dBm	-	-	Multilevel Sensor Report (01, 22, 00, E1)
→	09.07.2020	10:24:00	12	1	-	-55 dBm	-	-	Multilevel Sensor Report (01, 22, 00, E1)

Figure 5.5 interface graphique du sniffer

De plus à a minuit aussi le fichier log du système Z-Wave (`/var/log/z-way-server.log` est sauvegardé. Finalement, pour avoir un détail plus fin, si nécessaire, chaque fois que les fichiers ou les paquets changent, une copie du delta est aussi stocké.

Pour le log du système, nous nous retrouvons avec un dossier qui contient des log que nous transformons en format CSV, fichiers.csv. Le nom est sous la forme "aaaammjj.csv". L'étape suivante se divise en deux parties dépendamment du traitement qui va être effectué :

Partie 1 : Si nous voulons appliquer l'analyse des fenêtres temporelles,

Nous prendrons les fichiers "aaaammjj.csv", nous les diviserons par heure de la journée(hh). De plus, nous ajouterons un indicatif(i) pour connaître quel jour de la semaine nous sommes. Nous obtiendrons, ainsi, des fichiers de cette forme "aaaaammjj_hh_i.csv". Le traitement qui suit a été expliqué dans la partie approche.

Partie 2 : Si nous voulons analyser les valeurs émises par les dispositifs dans une plage horaire bien précise, nous procéderons, d'abord, au choix de la localité(l) du dispositif (dans quelle pièce de la maison, il se trouve.), sa dénomination, ainsi que sa typologie (la plupart des dispositifs de température donnent aussi l'humidité, sauf que, dans les logs, ce sont deux lignes différentes.). Nous ne considérerons pas l'ID du dispositif. Cela parce que le système change l'ID du composant : des fois des pannes de système déclenchent l'exclusion d'un dispositif et lors de la re - inclusion du même dispositif son ID change. Nous obtenons des fichiers de cette

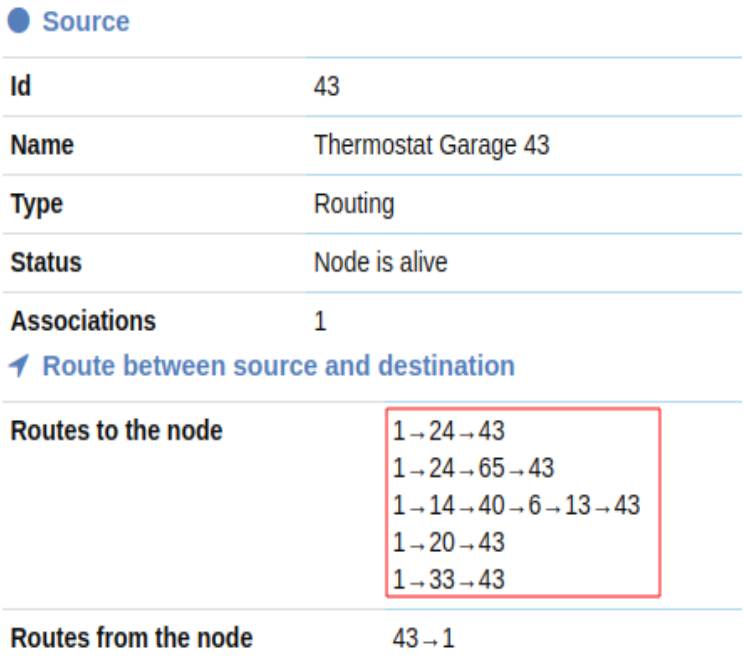


Figure 5.6 différentes routes possibles 43 (9 juillet)

forme "aaaammjj_l.csv" Ensuite, selon la plage horaire sur laquelle nous voulons effectuer nos tests, nous allons former nos vecteurs temporels de valeurs.

Enfin, nous allons créer des oracles pour valider l'exactitude du comportement des sous-systèmes. Par exemple, pour le sous-système de chauffage (thermostats et éléments chauffants) selon l'horaire programmé il est possible de définir des oracles. Pour notre cas, nous avons choisi de le faire manuellement et cela pour plus de précision. De ce fait, nous avons opté pour le mois de novembre et décembre (2019) où la température à l'extérieur est toujours basse. Nous avons convenu de 61 vecteurs de valeurs de température entre minuit et 5 h du matin. Ce choix est, bien entendu, en relation avec notre système. En effet, durant cette plage horaire, la courbe de tendance devrait toujours diminuer (ce qui n'est pas le cas). À noter, que nous ne pouvons pas savoir de quelle façon la température va diminuer pour une localité donnée. Le niveau idéal minime à attendre (et parfois attendu), prévu par le système est de 16 - 18 Celsius selon la pièce. Mais la température a l'extérieur, les autres pièces, les meubles et les personnes ont un impact. Donc la seule chose que nous pouvons faire, c'est de décider (oracle) si la diminution attendue est vraiment observée.

Cet oracle va être par la suite comparé aux classifications trouvées (croissance, décroissance, stabilité) par les deux méthodes de k-means et de calcul de la tendance (Trend).

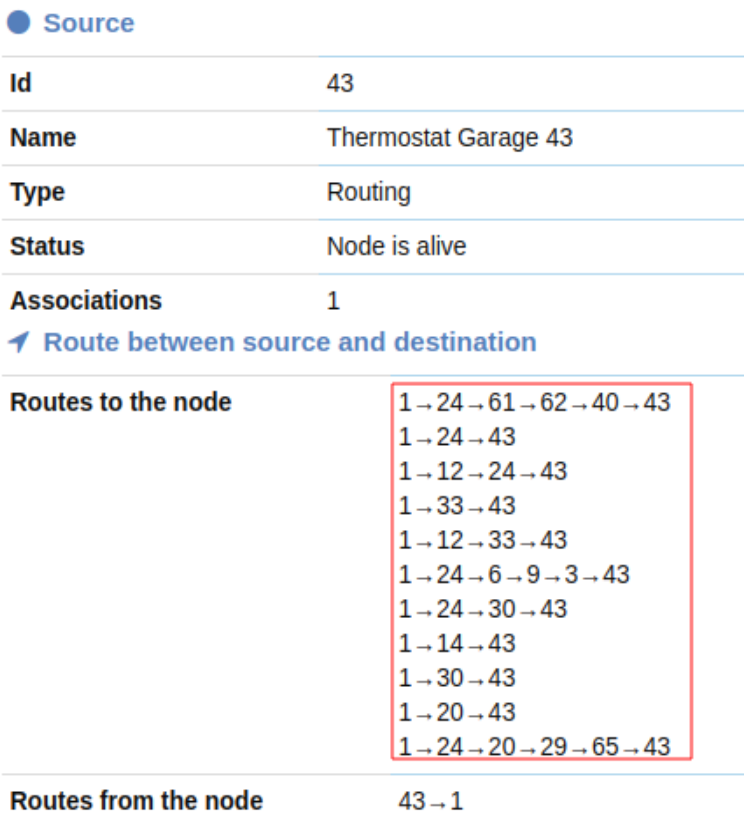


Figure 5.7 différentes routes possibles 43 (10 juillet)

5.2.1 Oracle

La figure 5.9 nous donne un aperçu de l'oracle émis. Comme nous pouvons le constater, le vecteur de valeurs est dans un ordre croissant de minuit à 5h du matin. De ce fait, pour calculer le degré de diminution(d), nous prendrons la différence entre la première et la dernière valeur. Voici l'hypothèse avancée :

- si d =< 0.5 : aucun changement 0
- si d>0.5 : la température a diminué 1
- si d négative : température a augmenté -1

Avec cette classification, nous remarquerons qu'il y a beaucoup d'incohérences. Voilà pourquoi il faudra, forcément, procéder au cas pas cas. Voici deux exemple :

Tableau 5.1 Exemple 1 oracle

Oracle	Diminution	Vecteur
1	-0.80	17.6, 17.6, 17.3, 17.3, 17.3, 17.3, 17.3, 17.3, 17.3, 18.1

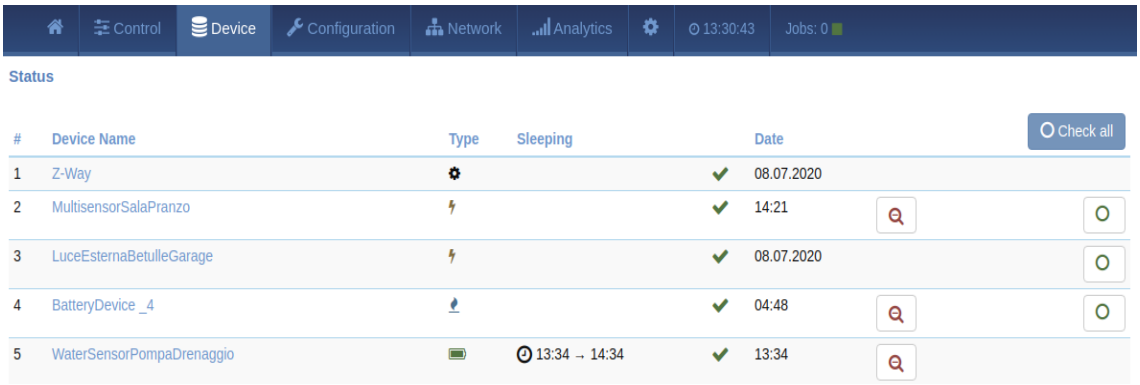


Figure 5.8 État des différents dispositifs

Exemple 1 tab 5.1 : Si nous nous fions à la classification énoncée au début, la valeur de l'oracle devrait être -1. Sauf qu'après-analyse des valeurs, nous remarquons que les valeurs sont correctes. Le système est programmée à commencer la diminution de la température vers 22 h et la température minimale de la pièce devrait être aux alentours de 17 degrés Celsius. De ce fait, des valeurs comprises entre 17 et 18 degrés Celsius devraient nous être très raisonnables.

Tableau 5.2 Exemple 2 oracle

Oracle	Diminution	Vecteur
1	0.69	22.8, 22.8, 22.3, 22.3, 22.3, 22.3, 22.3, 22.3, 22.3, 22.1, 22.1

Exemple 2 tab 5.2 : ici, "d" est positif mais les valeurs dans notre vecteurs sont trop élevées, par rapport aux valeurs attendues. Notons que nous acceptons une marge d'erreur de 1 degré. donc, ce cas représente un vrai négatif.

Par conséquent, il n'y pas de méthode fixe pour la création de notre oracle, c'est purement du cas par cas. Plusieurs facteurs entrent en jeu soit le comportement du système ou les attentes de l'utilisateur. Donc, nous nous sommes fiés à notre expertise pour faire la classification.

5.3 Description de notre système

le système est implémenté dans une maison qui comporte 2 niveau (sous-sol,premier étage), 15 pièces avec un total de 45 dispositifs de différents types. Nous avons principalement des captateurs de mouvements, des thermostats et des relais. Dans le tableau 5.3 nous identifions tous les capteurs selon leurs emplacement.

Les différents capteurs communiquent sur un réseau maillé comme le montre la figure 5.10

Oracle	Diminution	Valeurs									
0	0.5	18.9	18.9	18.4	18.4	18.4	18.4	18.4	18.4	18.4	18.4
1	-0.800000000000001	17.6	17.6	17.3	17.3	17.3	17.3	17.3	17.3	17.3	18.1
0	0.399999999999999	18.9	18.9	18.9	18.9	18.9	18.9	18.5	18.5	18.5	18.5
0	0.199999999999999	18.8	18.8	18.8	18.8	18.8	18.8	18.6	18.6	18.6	18.6
0	0.300000000000001	18.2	18.2	18.2	18.2	18.2	18.2	17.9	17.9	17.9	17.9
0	0.5	18.7	18.7	18.7	18.7	18.7	18.7	18.2	18.2	18.2	18.2
-1	0.800000000000001	18.1	18.1	18.1	18.1	18.1	18.1	17.3	17.3	17.3	17.3
0	0.399999999999999	23.7	23.7	23.7	23.7	23.7	23.7	23.3	23.3	23.3	23.3
0	0.5	18.4	18.4	18.4	18.4	18.4	18.4	17.9	17.9	17.9	17.9
-1	0.699999999999999	18	18	18	18	18	18	17.3	17.3	17.3	17.3
-1	3.8	23.7	23.7	23.7	23.7	23.7	19.9	19.9	19.9	19.9	19.9
-1	3.2	22.6	22.6	22.6	22.6	22.6	19.4	19.4	19.4	19.4	19.4
-1	3.3	23.3	23.3	23.3	23.3	23.3	20	20	20	20	20
1	-0.600000000000001	22.5	22.5	22.5	22.5	22.5	23.1	23.1	23.1	23.1	23.1
1	-0.5	21.6	21.6	21.6	21.6	21.6	22.1	22.1	22.1	22.1	22.1
0	0.400000000000002	22.6	22.6	22.6	22.6	22.2	22.2	22.2	22.2	22.2	22.2
1	-0.5	20.8	20.8	20.8	20.8	21.3	21.3	21.3	21.3	21.3	21.3
-1	2.1	21.9	21.9	21.9	21.9	21.9	19.8	19.8	19.8	19.8	19.8
	0.100000000000001	21.8	21.8	21.8	21.8	21.8	21.7	21.7	21.7	21.7	21.7

Figure 5.9 Oracle

nous pouvons voir qu'il y a différentes sortes de liens, dépendant de l'épaisseur de la ligne sur le graphe, plus c'est épais plus le lien est fort, nous avons aussi une description plus détaillée avec les routes possible pour chaque nœud. Notons aussi que nous disposons de plusieurs applications locales que nous pouvons exploiter notament on a (six thermostat virtuel, six dispositif fictif (utilisés pour s'interfacer avec MQTT) , cinq scène, 21 calendriers, neuf règles logique.

Au cours de ce projet, nous avons manipulé plusieurs versions de software. Les premières implémentations du système du système ont commencé en 2017, ou il y avait plusieurs fonctionnalités manquantes comme le sniffer, la table de routage détailler. À ce jour la version logicielle installée et fonctionnelle est V3.0.6 comme le montre la figure 5.11

5.4 Question de Recherche

Nous avons suivi les directives de Wohlin *et al.* [41] pour planifier et mener notre étude de cas [41]. L'objectif de notre étude de cas, en utilisant le GQM (Goal Question Metrics) élaboré par Basili *et al.* [42], a été définie comme suit : Analyser les capacités des différentes approches de détection des anomalies, afin de les comparer selon le type et la facilité d'utilisation, pour détecter les anomalies du point de vue du chercheur et du praticien, dans le contexte d'un

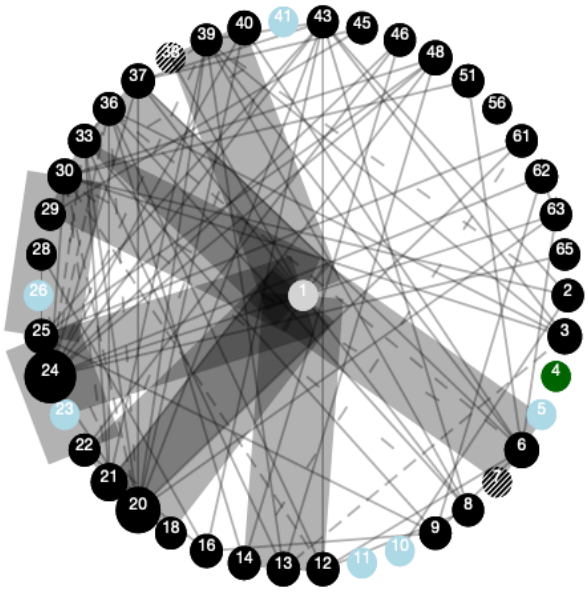


Figure 5.10 Réseau en maille des différents dispositifs

système domotique hybride WIFI Z-Wave .

Compte tenu de notre objectif de haut niveau et de l'approche proposée, y compris les différentes techniques, nous avons formulé et étudié les questions recherches suivantes :

QR1 : Dans quelle mesure l'approche proposée peut-elle détecter efficacement les anomalies discrètes et continues ?

Notre objectif est d'étudier et de vérifier si la base heuristique et la densité locale des séries temporelles aident à localiser cette famille d'anomalies.

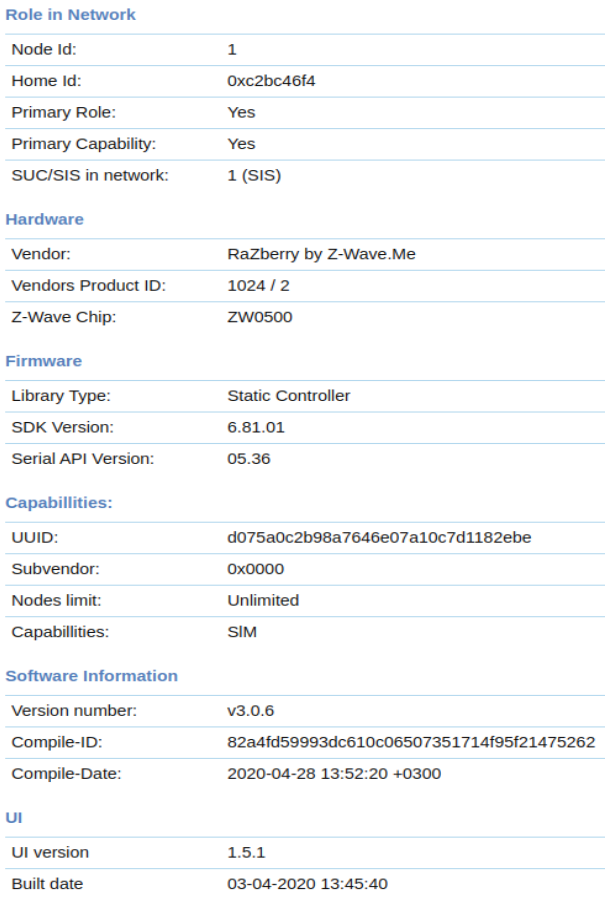
QR2 : Quelle est l'efficacité de l'approche proposée pour détecter les anomalies fantômes et anti-causales ?

Comme expliqués dans l'approche, les deux phénomènes sont imbriqués ; une anomalie fantôme entraîne souvent une anomalie anti-causale.

QR3 : Y a-t-il un accord entre les anomalies anti-causales détectées et le jugement de l'utilisateur ?

En effet, lorsqu'une anomalie anti-causale est détectée, impliquant un biais dans les observations, les tendances des mesures en accord avec l'algorithme de détection peuvent ou non être jugées par l'utilisateur.

Notre système de test est en place depuis deux ans. Il est à son état final, depuis la mi-2019, c'est-à-dire qu'aucun dispositif supplémentaire n'a été ajouté. Pour simplifier la définition des



Role in Network

Node Id:	1
Home Id:	0xc2bc46f4
Primary Role:	Yes
Primary Capability:	Yes
SUC/SIS in network:	1 (SIS)

Hardware

Vendor:	RaZberry by Z-Wave.Me
Vendors Product ID:	1024 / 2
Z-Wave Chip:	ZW0500

Firmware

Library Type:	Static Controller
SDK Version:	6.81.01
Serial API Version:	05.36

Capabillities:

UUID:	d075a0c2b98a7646e07a10c7d1182ebe
Subvendor:	0x0000
Nodes limit:	Unlimited
Capabillities:	SIM

Software Information

Version number:	v3.0.6
Compile-ID:	82a4fd59993dc610c06507351714f95f21475262
Compile-Date:	2020-04-28 13:52:20 +0300

UI

UI version	1.5.1
Built date	03-04-2020 13:45:40

Figure 5.11 information sur le contrôleur

algorithmes, le réglage et la validation des données, nous avons analysé et rapporté les données de novembre 2019 au début du printemps 2020. À cette époque de l'année (en particulier en novembre et décembre 2019), les températures extérieures sont inférieures à zéro et l'humidité est assez stable et faible, ce qui facilite la détection des anomalies anti-causales.

Tableau 5.3 Description des dispositifs

ID	Name	Place	Type	MovementType
1	Z-Way	Basement	Movable	High
2	MultisensorSalaPranzo	FirstFloor	Movable	Medium
3	LuceEsternaBetulleGarage	FirstFloor		
4	BatteryDevice			
5	WaterSensorPompaDrenaggio	Basement	Movable	High
6	SirenaAllarme	FirstFloor	Movable	High
7	Device			
8	DimmerTelevisione	FirstFloor		
9	ReleCameraMatrimoniale	FirstFloor		
10	PortaGarageSottosuolo	Basement		
11	BatteryDevice			
12	TermostatoZonaStufaSottosuolo	Basement		
14	ReleSoggiornoEPortaIngresso	FirstFloor		
16	DimmerBagnoPrimoPiano	FirstFloor		
18	DimmerCucilaIngresso	FirstFloor		
19	DimmerCucinaTavola	FirstFloor		
20	TermostatoZonaIngressoSottosuolo	Basement		
21	SwitchLuceEsternoIngressoEGarage	FirstFloor		
22	HomeWaterValveStJoseph	Basement		
23	Porta Ingresso - Main door	FirstFloor		
24	LampadaSteloDimmedSalaPranzo	FirstFloor	Movable	High
25	SwitchLevitonPortabile			
26	PortaIngressoGarageSoggiorno	FirstFloor		
28	MasterBedroomMultisensor	FirstFloor	Movable	Medium
29	MultisensorStudioCameterra	FirstFloor		
30	Boiler - Water Tank Heater	Basement		
33	FirsFloorPrimoPiano	FirstFloor	Movable	Medium
36	MultisensorSoggiorno	FirstFloor	Movable	Low
37	BPPMultisensor	FirstFloor	Movable	Medium
38	MainsDevice			
39	MultisensorGarage	FirstFloor	Movable	Low
40	LampadaSteloCameraDaPochiSoldi	FirstFloor	Movable	
41	Porta Cucine Giardino	FirstFloor		
42	MainsDevice			
43	Thermostat Garage	FirstFloor		
45	ReleBagnoPP	FirstFloor		
46	ReleCamerettaStudio	FirstFloor		
48	SwitchLuceEsternoCucine	FirstFloor		
51	DimmerGarage	FirstFloor		
56	Temporary Socket Device		Movable	High
61	Temporary Dev 2	Basement	Movable	High
62	ReleSalaPranzoCucina	FirstFloor		
63	ZonaStufaSottosuolo	Basement		
65	Pompa Piscin	FirstFloor		

CHAPITRE 6 RÉSULTAT

Dans ce chapitre, nous présentons et analysons les résultats obtenus par les différentes méthodes de détection développées.

Nous répondrons, également, aux trois questions de recherche qui visent à vérifier la faisabilité de détecter les anomalies dont nous avons donné les définitions dans les chapitres précédents.

Nous avons observé que les activités journalières n'ont pas toujours un comportement régulier. En effet, nous avons trouvé des pics d'activités difficiles à comprendre *e.g.,* pendant la nuit. Nous avons, donc, commencé par effectuer une analyse non supervisée de détection des anomalies avec des fenêtres fixes, comme expliqué, précédemment, dans l'approche. La figure 6.1 indique la répartition du nombre d'anomalies (*i.e.,* les valeurs aberrantes pics d'activités) par heure et par jour de la semaine. Nous ne montrons que trois jours ; un comportement similaire a été observé pour les autres jours. Il ressort clairement de la figure que nous avons, en moyenne, deux ou trois anomalies. Ces chiffres peuvent atteindre les 30 anomalies ou plus, par heure, chaque jour. Il convient de mentionner que dans la figure 6.1, une anomalie est considérée comme telle, si et seulement si, les trois experts ont classé la fenêtre temporelle comme anormale.

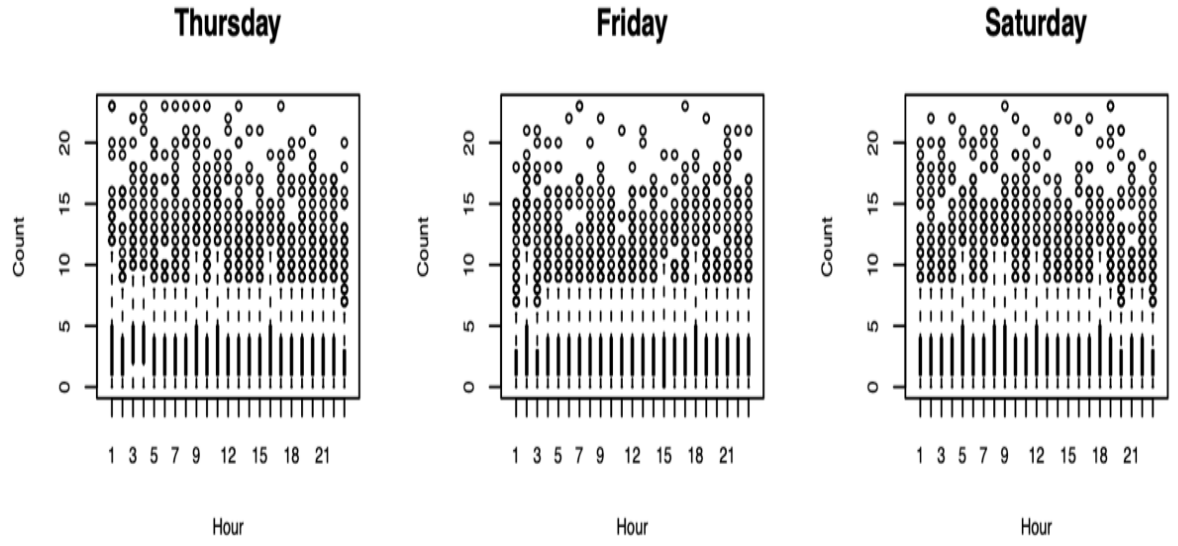


Figure 6.1 Résultats de détection des anomalies en se basant sur les trois experts - statistiques sur cinq mois

En effet, l'observation des données nous a permis de trouver plusieurs mesures incohérentes,

soit avec les lois de la physique, soit avec la logique de programmation du système. Pour exemple, il est évident que nous ne pouvons pas avoir une température de 1.3M Celsius, comme nous pouvons voir dans la figure 4.6, ni de -470 Celsius. Aussi, il est envisageable que les délais entre les causes et les effets soient limités, mais, il nous est déjà arrivé d'observer des retards de minutes. Une première hypothèse a été faite d'un lien entre les activités du système et, les erreurs et les retards. Pour valider cette hypothèse à partir des résultats de l'analyse des trois experts à fenêtre fixe (figure 6.1) nous avons émis l'hypothèse qu'il y a un lien entre les pics d'activités et les autres anomalies.

L'idée était de rechercher les anomalies en correspondance avec les pics d'activités. Malheureusement, à date, nous n'avons pas trouvé un lien fort entre les pics d'activités et les anomalies. Par exemple, nous pourrions spéculer qu'il existe un lien entre les retards (*e.g.,* mouvement détecté et lumière allumée). Cependant, si nous avons à calculer la corrélation (*e.g.,* pourcentage d'anomalies versus les retards en seconds) nous pouvons constater qu'elle est plutôt faible, en dessous de 30%. Nous avons, donc, décidé de poursuivre les analyses, pour l'instant, sans considérer les pics d'activités.

Dans les paragraphes suivants, nous allons exposer les diffèrentes analyses appliquées et les résultats obtenus sans considérer les pics d'activité.

6.1 QR1 : Dans quelle mesure l'approche proposée peut-elle détecter efficacement les anomalies discrètes et continues

Nous avons appliqué différentes analyses en tenant compte de l'analyse exploratoire des données, des connaissances, des statistiques et des propriétés locales. L'analyse exploratoire des données permet de les visualiser et de les inspecter rapidement. Nous avons observé des situations où il est évident qu'une anomalie s'est produite.

Tableau 6.1 Températures : statistique descriptives.

Dispositif	Min.	25 Perc.	Median	75 Perc.	Max.	IQR
20	-3206	16	17	17	1116068	1
37	14	17	18	21	25	4
29	-17	16	17	18	21	2
2	14	17	18	19	22	2
28	-8	17	17	18	342	1
36	15	18	18	19	29	1
27	17	19	19	20	22	1
33	10	17	18	18	3720	1
4	10	12	12	12	17	0

6.1.1 Technique Standard : 1.5* IQR

Le point positif est que l'analyse est rapide et très intuitive. Plus formellement, une anomalie discrète ou continue peut être défini par rapport à la distribution globale des données ou par rapport au comportement local de la série temporelle. Les valeurs aberrantes globales sont souvent définies comme des valeurs supérieures à 1,5 de l'intervalle inter-quartile au-dessus du 75e centile, et sont, donc, définies par rapport à l'ensemble de l'historique de la distribution des données. Nous avons utilisé cette définition pour analyser la température et l'humidité des valeurs émises par les capteurs implanter dans notre système.

Dans les tableaux 6.3 et 6.4 nous avons les résultats de la classification en nous basant sur la définition des valeurs aberrantes [1] par rapport à une analyse de toutes les données propres au dispositif. Mais avant, nous avons effectué des statistiques générales comme les montres les tableaux 6.1 et 6.2 pour avoir une vue d'ensemble des valeurs et pour calculer les quartiles inférieur et supérieur qui sont nos repère de classification.

Pour le tableau 6.3 qui représente des capteurs de température. Si nous prenons l'exemple de l'élément 20 nous avons un total de 14659 valeurs dont 39 qui sont des valeurs aberrantes inférieures qui appartiennent à l'intervalle de valeurs $Q(25perc : 16) - 1,5 * IQR(1)$, 31 Valeurs aberrantes supérieures incluses dans l'intervalle $Q(75perc : 17) + 1,5 * IQR(1)$, 4 valeurs valeur inférieure à 5 degrés Celsius et 29 valeurs supérieures à 40 degrés Celsius. Nous avons appliqué ces mêmes règles sur tous les dispositifs de température.

Tableau 6.2 Humidité : statistique descriptives.

Dispositif	Min.	25 Perc.	Median	75 Perc.	Max.	IQR
39	35	46	52	60	84	14
28	34	49	55	66	88	17
37	28	46	52	58	98	12
2	30	41	45	52	76	11
33	32	42	46	55	100	13
29	44	51	54	66	79	15
63	0	38	43	51	67	13

Avec le tableau 6.4 qui représente les capteurs d'humidité nous avons eu la même approche, mais les valeurs de comparaison changent. Si nous penons l'exemple de l'élément 39 nous avons un total de 11938 valeurs dont 0 qui sont des valeurs aberrantes inférieures, 0 valeur aberrante supérieure, 0 valeur inférieure à 10 et 11 valeurs supérieures à 80.

1. `https://www.itl.nist.gov/div898/handbook/prc/section1/prc16.htm`

Tableau 6.3 classification par dispositif des valeurs de la température

Device	Total value	Lower outlier	Upper outlier	<5C	>40C
20	14659	39	31	4	29
37	10351	0	0	0	0
29	12985	9	0	9	0
2	13238	0	0	0	0
28	13077	12	27	5	7
36	20280	0	306	0	0
27	4159	0	0	0	0
4	1687	160	196	0	0
33	11783	14	146	0	2

Tableau 6.4 classification par dispositif des valeurs de l'humidité

Device	Total value	Lower outlier	Upper outlier	<10	>80
39	11938	0	0	0	11
28	12871	0	0	0	16
37	10108	0	7	0	138
2	11615	0	0	0	0
33	11411	0	14	0	119
29	12564	0	0	0	0
63	13453	0	0	1	0

6.1.2 Technique : "Local Outlier Factor"

Nous avons traité les séries temporelles en utilisant la détection de valeurs aberrantes non supervisée de Python avec l'algorithme "local outlier factor" implémenté dans "sklearn.neighbors LocalOutl ierFactor" réglé par défaut (prend des échantillons de 20 valeurs du voisinage pour la prédiction). Pour chaque série temporelle, nous avons localisé des anomalies discrètes, et chaque fois que des points consécutifs étaient classés comme anomalies, nous avons appliqué une détection d'anomalies continue. Les résultats du traitement pour les capteurs de température sont présentés dans la figure 6.2. Cette analyse a été effectuée sur 9 capteurs de température dont 2 ne présentant aucune anomalie et les 7 autres un mélange entre des anomalies discrètes et continues. L'axe des X indique les ID des nœuds. L'axe des Y indique les nombres discrets, continues et la somme totale des deux types d'anomalies.

Dans l'ensemble, sur 212 anomalies de température identifiées, 102 étaient des valeurs supérieures aux chiffres prévus pour la saison, dont beaucoup (51) étaient supérieures à 40 degrés Celsius.

Pour les capteurs d'humidité, les résultats sont présentés dans la figure 6.3. D'après les gra-

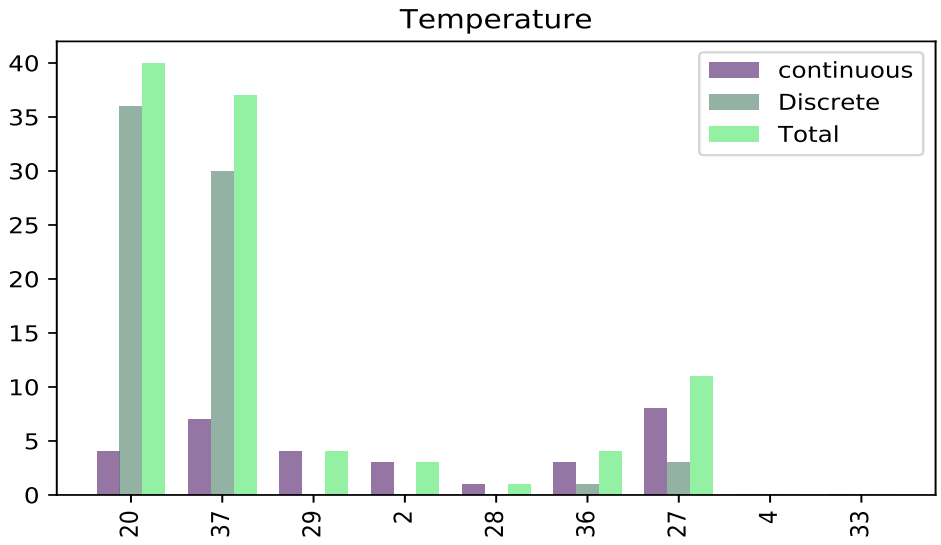


Figure 6.2 Nombre d'anomalies de température par dispositif (statistique sur deux mois "novembre - décembre")

phiques, nous pouvons voir que certains capteurs présentent plus d'anomalies que d'autres. Cela peut indiquer un problème de communication réseau avec ces appareils, la présence d'obstacles, de chemins de communication complexes ou d'un dysfonctionnement de l'appareil.

6.1.3 Localisation des anomalies discrètes et continues basée sur la connaissance

De nombreux système cyber-physiques traitent d'objets et de mesures physiques limités par des lois physiques. Ces connaissances aident à localiser les anomalies, tenir compte de la température (ou de l'humidité) des capteurs. Pour de nombreux appareils, à l'intérieur ou à l'extérieur du bâtiment, les valeurs des variables mesurées ont des limites connues et les plages de fonctionnement sont, à priori, connues. Par conséquent, une méthode rapide consiste à rechercher les valeurs de TS à l'extérieur des fourchettes attendues. Par exemple, toute température au-dessus de 40 (ou moins de zéro) degrés Celsius peut être considérée comme une anomalie.

Dans l'ensemble, nous concluons que la connaissance et la méthode des facteurs aberrants

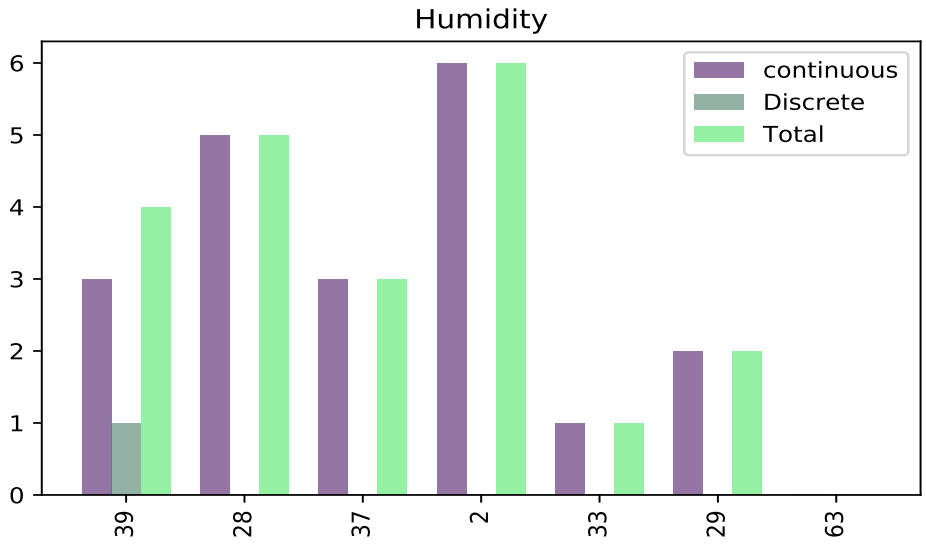


Figure 6.3 Nombre d'anomalies d'humidité par dispositif (statistique sur deux mois "novembre - décembre").

locaux sont suffisantes pour localiser les anomalies ponctuelles et d'intervalle.

6.2 QR2 : Quelle est l'efficacité de l'approche proposée pour détecter les anomalies fantômes et anti-causales

Il est important de rappeler que les fantômes, par définition, ne peuvent pas être détectés par la seule analyse des logs. Ils correspondent à une action qui a, apparemment, été effectuée, mais qui en réalité, ne s'est jamais produite. Il peut y avoir plusieurs raisons à cela. Une des raisons pourrait être une congestion du réseau.

Empiriquement, nous savions que parfois la température ne diminuerait (ou n'augmenterait) pas comme prévu. Nous avons, donc, émis l'hypothèse de l'existence d'événements fantômes. Par conséquent, nous avons procédé à l'application de deux techniques : le clustering et la trend. En fonction de l'horaire et du dispositif de chauffage, nous avons surveillé l'intervalle de temps entre 21h00 (22 et 22h30) et 6h00 du matin. A 21h00 (22 et 22h30), une commande est envoyée pour éteindre le chauffage des appareils 45, 14 et 62, respectivement. En conséquence, les capteurs de température 2, 33, 36 et 37 doivent signaler une diminution de la température.

Notez bien que les capteurs 2, 33 et 36 sont dans la même pièce.

6.2.1 Clustering

La logique suggère que la température augmente ou diminue ou reste à peu près constante. Donc si nous regroupons les comportements des capteurs de température nous nous attendons à obtenir trois groupes. Nous avons extrait, de chaque capteur ST, l'intervalle correspondant

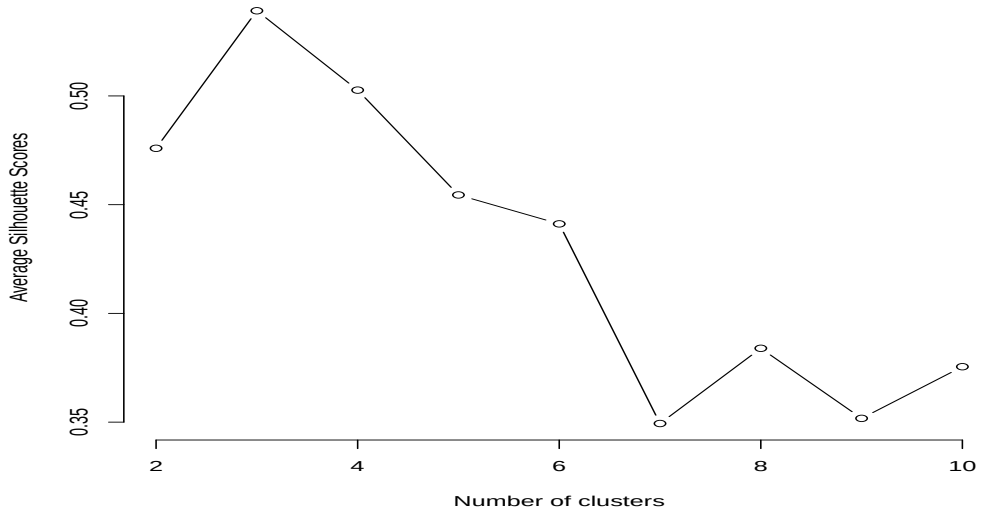


Figure 6.4 cluster Kmeans

à la nuit et à l'heure à laquelle le chauffage devrait être éteint (disons 21 :00/22 :00/22 :30 à 6 heures du matin). Ensuite, pour chaque capteur, nous avons utilisé l'algorithme DTW(Dynamic Time Warping) qui permet de calculer la similitude entre deux série temporelle de longueurs différentes et cela pour calculer toutes les paires de distances pour chacun des sous-ST des valeurs de la température. Enfin, nous avons appliqué deux : K-means est une technique non supervisé qui vise à partitionner n observations en k groupes selon leur similarité.

Le résultat optimal est en général le plus petit nombre de claustres qui a le score "silhouette" [43] le plus élevé comme nous le montre la silhouette, voir la figure 6.4.

Le cluster agglomeratif, connu aussi sous le nom d'Agnes [44],définie comme cluster hiérarchique utiliser pour regrouper des valeurs en fonction de leurs similitudes. Une fois exécuter, il génère un dendrogramme comme la figure 6.5.

Comme nous pouvions nous y attendre en appliquant les deux méthodes, nous avons conclu que le nombre optimal de clusters est de trois : température stable, croissante et décroissante.

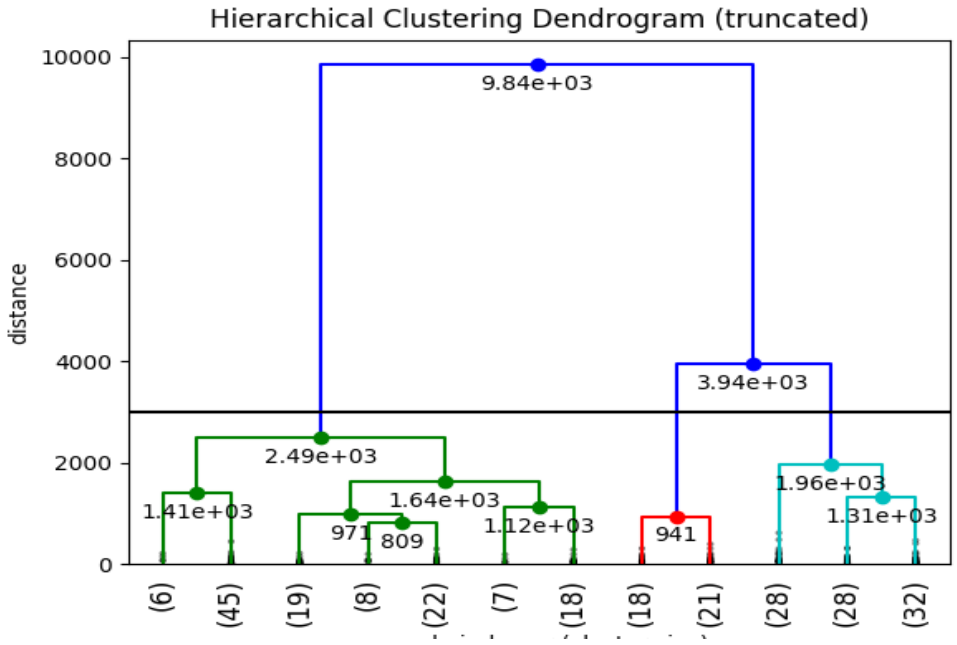


Figure 6.5 agnes

6.2.2 Trend

Le clustering et le DTW considèrent globalement deux par deux la similarité des intervalles des TS. Une approche différente, guidée par la connaissance, imposerait une tendance à la baisse de la température. Nous avons donc appliqué la statistique de la pente de Sen, la statistique de Lanzante [39] et le test de tendance de Cox et Stuart [40]. Nous avons, également, appliqué une régression linéaire simple et, par la suite, calculé le coefficient de pente. Sur cet ensemble de données, en particulier, nous avons constaté que la statistique de Lanzante ne permettait souvent pas de détecter la présence ou l'absence de la tendance. Au contraire, les tests de Cox, Sen et Stuart se sont avérés plus robustes pour identifier la présence d'une tendance. Cependant, les tests ne signalent pas la valeur de la tendance. En fin de compte, nous décidons de comparer avec l'oracle manuel les résultats de la seule statistique de la pente de Sen.

6.2.3 QR3 : Y a-t-il un accord entre les anomalies anti-causales détectées et le jugement de l'utilisateur

Du 1er novembre au 31 décembre, pour trois chambres (location 7, location 2 et 3) nous avons extrait les 61 séries temporelles des valeurs de la température durant la nuit. Nous les avons étiquetés comme étant en hausse, en baisse et constante. Enfin, nous avons comparé l'oracle

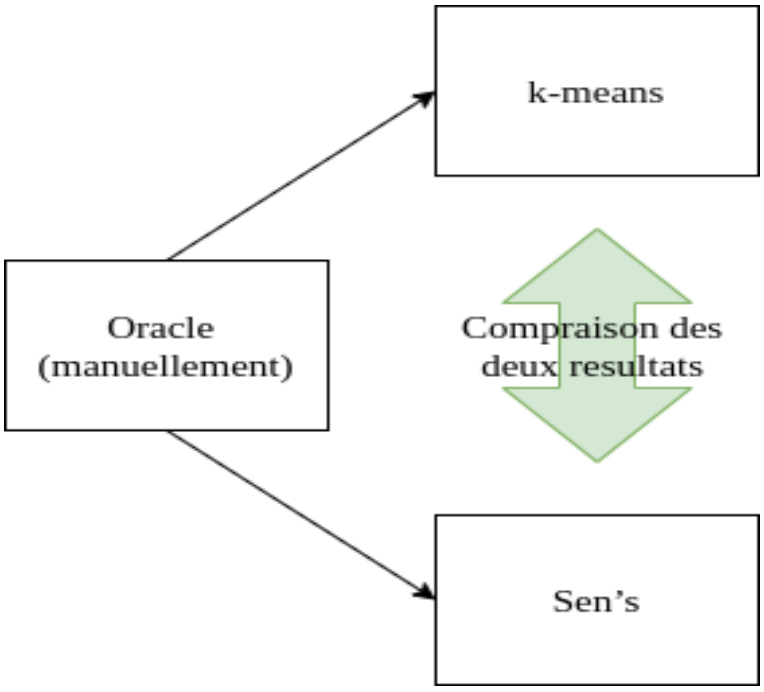


Figure 6.6 comparions de l'oracle avec K-means et Sen's

produit manuellement avec la cluster K-means 6.5. Nous avons constaté que le cluster 2 avait le comportement attendu. Le tableau 6.6 montre la matrice de confusion ; le test Chi-square rejette l'hypothèse nulle selon laquelle les données sont dues au hasard avec une valeur p de 0,02.

Tableau 6.5 Cluster K-means

k-means	Vecteur
2(croissante)	20.8 20.8 20.8 20.8 21.3 21.3 21.3 21.3 21.3 21.3
3 (décroissante)	21.9 21.1 21.1 21.1 21.1 21.1 21.1 21.1 18.8 18.8 18.8
1(stable))	18.9 18.9 18.4 18.4 18.4 18.4 18.4 18.4 18.4 18.4

La précision est d'environ 62 %. Il faut cependant noter que certaines valeurs mesurées n'ont pas été identifiées comme décroissantes par l'annotateur humain car la diminution de la température n'a pas été jugée suffisante. À notre grande surprise, nous avons découvert, non seulement, une tendance non décroissante, mais aussi une série temporelle, qui montre une légère augmentation de la température. Pour les nuits où l'anomalie a été observée, nous avons inspecté manuellement les logs et vérifié que la commande d'arrêt du chauffage avait été envoyée, acquittée et qu'aucune erreur ne suivait, immédiatement, l'acquittement de la commande de séquence.

Tableau 6.6 Table de confusion de la classification entre l'oracle et K-means des températures en diminution - emplacement 7

	Diminution	Non-diminution
Diminution	27	19
Non-diminution	4	11

Tableau 6.7 Table de confusion de la classification de la tendance entre l'oracle et la sen's des températures en diminution - emplacement 7

	Diminution	Non-diminution
Diminution	40	0
Non-diminution	5	16

De la même manière, nous avons comparé l'analyse de la pente de Sen's avec l'oracle manuel pour la classification des tendances décroissantes. Les résultats sont présentés dans le tableau 6.7. La pente de Sen's a une précision beaucoup plus élevée, atteignant 91 %, et le test du Chi-square renvoie une valeur p inférieure à 0,00001.

Globalement, nous avons conclu que l'analyse des clusters et des tendances associées à la connaissance permettent d'identifier les anomalies anti-causales. Par conséquence, elles donnent lieu à une localisation implicite des régions temporelles dans lesquelles des anomalies fantômes se produisent dans les logs.

Cependant, d'après nos données, le test de Sen's et la pente de régression linéaire ont donné de bien meilleurs résultats. Ils présentent, également, l'avantage de pouvoir être facilement mis en œuvre, quasiment en temps réel, par un observateur. L'idée, comme indiquée, est similaire à celle de `fail2ban`. Des mesures s'accumulent, disons 4 - 5 mesures (c'est-à-dire 20 - 30 min après l'événement on/off attendu) de la pente de régression. En cas de désaccord avec le comportement attendu - send/resend la commande on/off.

Les analyses effectuées jusqu'à présent permettent d'identifier le phénomène, mais n'aident pas à en comprendre la cause profonde de ces questions. Un interrupteur est soit allumé, soit éteint, mais un variateur d'intensité passe entre différents niveaux avant d'atteindre la valeur cible. Les valeurs intermédiaires sont indiquées dans les fichiers logs. Nous avons, donc, étudié le décalage entre un mouvement et la lumière. En figure 4.5, nous observons que cette différence de temps a une grande variabilité. Elle peut aller jusqu'à 500 sec. Les méthodes de "local outlier factor" ont permis d'identifier environ 61 valeurs aberrantes. Les décalages de plus de 7 secondes ont été classés comme anormaux.

Les résultats préliminaires confirment une corrélation entre un long décalage dans le temps,

un mauvais RSSI et une augmentation de la retransmission. Il semble que l'un des facteurs critiques soit les murs et les interférences. En effet, les murs et la distance expliquent bien le RSSI en utilisant le modèle de régression linéaire. Cependant, cela n'explique pas pourquoi il y a des commandes manquantes dans les fichiers logs, des fantômes, des anomalies anti-causales ou des aberrations extrêmes.

Un phénomène observé dernièrement qui peut être la cause de notre problème est la synchronisation entre notre intercepteur (contrôleur de chauffage) et les valeurs réelles observées à travers le thermostat.

Pour illustrer davantage ce point, considérons Fig 6.7 : le rapport quotidien pour le capteur 36, avec les commandes "ON"/"OFF" envoyées à l'interrupteur lié au thermostat virtuel (interrupteur 14). Les moments de commutation sont marqués par des points bleus. Pour cette pièce, la température doit rester à 17 °C la nuit et à 20 °C le jour. Il est fixé à 20 entre 12 et 14 ; puis entre 19 et 22 heures. Nous constatons une anomalie ponctuelle vers 2 heures du matin : il y a un événement étrange. Le chauffage n'était pas censé se mettre en marche. Vers midi, nous observons une anomalie d'intervalle. Tous les autres capteurs de température situés dans la même pièce ont signalé une température proche de 20 °C et ne dépassant pas 22 °C. Ici, nous observons également des événements fantômes : il manque deux commandes pour allumer puis éteindre l'appareil 14 avant l'augmentation de la température. Le dispositif 14 contrôle l'élément chauffant et doit donc être "allumé" afin d'augmenter la température, mais la dernière commande était "éteinte".

6.3 Parcours et tables de voisinage

Le système Z-Wave représente le maillage des appareils dans de fichiers format JSON. En effet, il décrit la notion de voisinage, c'est-à-dire quels dispositifs nous pouvons atteindre (avec un seul hop) à partir d'un autre élément donné. La figure 6.8, est le résultat de l'élaboration du fichier JSON pour un jour donné. Nous remarquons que le nœud 1, c.à.d, le contrôleur ne peut pas, par exemple, communiquer avec le nœud 5, mais il peut atteindre le nœud 3, qui lui-même a un lien avec le nœud 5. Donc entre 1 et 5, il pourra soit passer par 3 soit choisir un parcours différent.

Nous avons émis l'hypothèse que, à l'aide de la table de voisinage, nous pourrons avoir accès aux routes possibles. Ce travail a été effectué avant que l'option de sniffing soit possible sur Z-Wave. Donc, nous nous sommes dits que, grâce à la table de voisinage, nous allons essayer de détecter les différents chemins possibles. Nous avons commencé par extraire tous les nœuds et tous les éléments auxquels ils étaient reliés via le fichier JSON pour le transformer en format

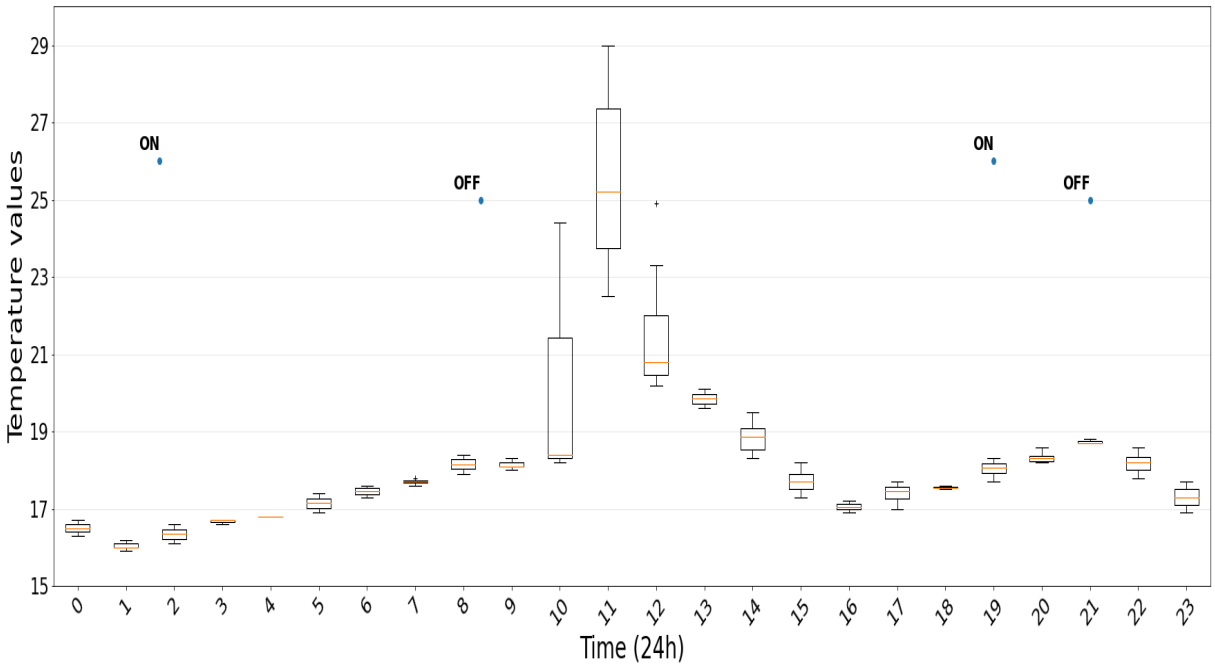


Figure 6.7 activité de la journée 2020-01-09 thermostat virtuel 36, interrupteur 14

XML pour simplifier l'analyse.

Après une analyse de plusieurs semaines nous sommes convaincus que les informations de voisinage étaient constant c.-à-d le routage était tout à fait indépendant du voisinage. La matrice de voisinage était toujours la même sauf si nous ajoutions ou retirions des dispositifs. Nous nous sommes rendu compte que les liaisons entre les dispositifs étaient toujours les mêmes, donc, aucun changement dans la table de voisinage. C'est comme si cette dernière n'avait été affectée qu'une seule fois lors de la création du système et que les routes changeaient indépendamment de ce tableau. Il est donc probable que les tables de routage sont modifiées selon la mesure de RSSI et le trafique des différents parcours.

```
#souce:Dest
1:[1, 3, 6, 7, 8, 9, 14, 16, 21, 22, 24, 25, 28, 29, 30, 42, 64]
2:[9, 10, 13, 14, 23, 26, 30, 31, 32, 46, 65]
3:[5, 6, 10, 12, 13, 18, 19, 20, 22, 23, 24, 25, 26, 27, 29, 30, 31, 32, 36, 39, 45, 46, 48, 51, 63, 65, 67]
5:[3, 6, 8, 12, 13, 14, 18, 20, 36, 48]
6:[1, 3, 5, 8, 10, 12, 13, 14, 16, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 36, 39, 45,
46, 48, 63, 65, 67]
7:[ ]
8:[5, 6, 12, 13, 18, 19, 20, 22, 23, 24, 25, 26, 27, 29, 31, 32, 36, 39, 45, 46, 48, 51, 67]
9:[2, 10, 12, 13, 18, 19, 20, 23, 24, 25, 26, 27, 30, 31, 32, 36, 39, 45, 46, 48, 51, 63, 65, 67]
10:[2, 3, 6, 9, 12, 13, 18, 19, 20, 21, 22, 24, 25, 39, 63]
11:[ ]
12:[3, 5, 6, 8, 9, 10, 13, 14, 16, 18, 19, 20, 21, 22, 24, 25, 26, 27, 28, 30, 31, 32, 36, 45, 46, 48, 51, 63, 65, 67]
13:[2, 3, 5, 6, 8, 9, 10, 12, 14, 16, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 36, 39, 45, 46, 48,
51, 63, 65, 67]
14:[2, 5, 6, 12, 13, 18, 19, 22, 23, 24, 25, 26, 28, 29, 30, 31, 32, 36, 39, 45, 46, 48, 51, 67]
16:[6, 12, 13, 18, 19, 20, 23, 24, 27, 30, 31, 32, 36, 45, 46, 48, 63, 67]
18:[3, 5, 6, 8, 9, 10, 12, 13, 14, 16, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 36, 39, 45, 46, 48, 51,
63, 67]
19:[3, 6, 8, 9, 10, 12, 13, 14, 16, 18, 20, 21, 22, 23, 24, 25, 27, 29, 31, 32, 36, 39, 45, 46, 48, 63, 65, 67]
20:[3, 5, 6, 8, 9, 10, 12, 13, 16, 18, 19, 21, 22, 23, 24, 25, 26, 27, 29, 30, 31, 32, 36, 45, 46, 48, 63, 67]
21:[6, 10, 12, 13, 18, 19, 20, 23, 24, 25, 26, 27, 31, 32, 36, 39, 45, 46, 48, 51, 63, 65, 67]
22:[1, 3, 6, 8, 10, 12, 13, 14, 18, 19, 20, 23, 24, 25, 26, 27, 28, 29, 31, 32, 36, 45, 46, 48, 63, 67]
23:[2, 3, 6, 8, 9, 13, 14, 16, 18, 19, 20, 21, 22, 24, 28, 36, 45, 46, 48, 51, 67]
24:[1, 3, 6, 8, 9, 10, 12, 13, 14, 16, 18, 19, 20, 21, 22, 23, 26, 27, 28, 29, 30, 31, 32, 36, 39, 45, 46, 48, 51,
63, 65, 67]
25:[1, 3, 6, 8, 9, 10, 12, 13, 14, 18, 19, 20, 21, 22, 27, 45, 46, 63]
26:[2, 3, 6, 8, 9, 12, 13, 14, 18, 20, 21, 22, 24, 36, 45, 48, 51, 67]
27:[3, 6, 8, 9, 12, 13, 16, 18, 19, 20, 21, 22, 24, 25, 28, 29, 36, 45, 46, 48, 63, 67]
28:[1, 6, 12, 13, 14, 18, 22, 23, 24, 27, 45, 46, 48, 63, 67]
29:[1, 3, 6, 8, 13, 14, 18, 19, 20, 22, 24, 27, 45, 46, 48, 65, 67]
30:[1, 2, 3, 6, 9, 12, 13, 14, 16, 18, 20, 24, 32, 36, 45, 48, 63, 67]
31:[2, 3, 6, 8, 9, 12, 13, 14, 16, 18, 19, 20, 21, 22, 24, 36, 39, 45, 46, 48, 51, 65, 67]
32:[2, 3, 6, 8, 9, 12, 13, 14, 16, 18, 19, 20, 21, 22, 24, 30, 36, 39, 46, 48, 51, 65, 67]
36:[3, 5, 6, 8, 9, 12, 13, 14, 16, 18, 19, 20, 21, 22, 23, 24, 26, 27, 30, 31, 32, 39, 46, 48, 51]
39:[3, 6, 8, 9, 10, 13, 14, 18, 19, 21, 24, 31, 32, 36, 45, 46, 51]
42:[]
45:[3, 6, 8, 9, 12, 13, 14, 16, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 39, 46, 48, 63, 65, 67]
46:[2, 3, 6, 8, 9, 12, 13, 14, 16, 18, 19, 20, 21, 22, 23, 24, 25, 27, 28, 29, 31, 32, 36, 39, 45, 48, 51, 63, 65, 67]
48:[3, 5, 6, 8, 9, 12, 13, 14, 16, 18, 19, 20, 21, 22, 23, 24, 26, 27, 28, 29, 30, 31, 32, 36, 45, 46, 51, 65]
51:[3, 8, 9, 12, 13, 14, 18, 21, 23, 24, 26, 31, 32, 36, 39, 46, 48, 65, 67]
63:[3, 6, 9, 10, 12, 13, 16, 18, 19, 20, 21, 22, 24, 25, 27, 28, 30, 45, 46]
64:[ ]
65:[2, 3, 6, 9, 12, 13, 19, 21, 24, 29, 31, 32, 45, 46, 48, 51]
67:[3, 6, 8, 9, 12, 13, 14, 16, 18, 19, 20, 21, 22, 23, 24, 26, 27, 28, 29, 30, 31, 32, 45, 46, 51]
```

Figure 6.8 Extraction de la table de voisinage des nœuds

CHAPITRE 7 CONCLUSION

Dans ce chapitre final, nous allons présenter un bref sommaire des résultats obtenus,les limitations auxquelles nous avons été confrontées ainsi que le travail futur envisagé.

7.1 Synthèse des travaux

Notre expérience nous montre que cette catégorie de systèmes n'est pas encore complètement prête pour le grand public. L'utilisateur s'attend à un système beaucoup plus facile à gérer [1], à configurer [2], à évoluer [3], et à faire fonctionner. En effet, la qualité, la robustesse, la configurabilité, la maintenance et l'évolutivité des logiciels sont des problèmes majeurs.

Nous pouvons dire que le système Z-Wave est un sous-produit du framework et des applications spécifiques de Z-Wave.me.

Nous avons également examiné et testé deux autres frameworks populaires : OpenHAB [4] et Domoticz [5]. Les deux ont des avantages et des inconvénients. Néanmoins, Z-Wave.me dispose d'un ensemble d'applications plus riche et d'une meilleure interface utilisateur pour les non-experts, spécifiquement pour les utilisateurs non avertis en matière de technologie. OpenHAB est plus orienté programmeur et plus flexible. Il peut s'adapter à presque tous les types d'appareils, mais son approche reste très complexe. Un utilisateur non averti aurait beaucoup de difficultés à l'utiliser.

Les solutions de cloud computing, *e.g.,* google home, ne sont pas à l'abri des problèmes également. Programmer des scènes impliquant différents appareils, horaires et actions risque d'être plus compliqué que d'effectuer les mêmes tâches avec les récents moteurs d'automatisation de Z-Wave.me. En plus des problèmes de confidentialité et les solutions de cloud computing échouent en l'absence de connexion Internet ce qui le rend très dépendant et donc non fiable.

Dans ce mémoire, nous avons présenté une approche automatique pour détecter les anomalies dans les systèmes IOT et cyber-physique. Cela en utilisant des séries temporelles et des informations extraites des fichiers logs. Nous avons, d'autant plus, introduit les notions d'anomalie fantôme et d'anomalie anti-causale. Nous avons également étudié les concepts d'anomalies discrètes et d'anomalies continues.

1. https://forum.z-wave.me/viewtopic.php?t=32804
2. https://forum.z-wave.me/viewtopic.php?t=20908
3. https://forum.z-wave.me/viewtopic.php?t=32860
4. https://www.openhab.org/
5. https://www.domoticz.com/

Nous avons, à la fois, défini des analyses spécifiques (*e.g.,* analyse en se basant sur la connaissance) et adapté des analyses tirées soit de la statistique (*e.g.,* définition de points aberrants) soit de l'apprentissage automatique (*e.g.,* local outliers factor).

De plus, pour ce qui est des anomalies fantôme et anti-causales, nous avons proposé des nouvelles techniques de localisation et de détection.

Nous avons appliqué l'approche proposée à un système cyber-physique du monde réel c'est-à-dire un système de domotique. Le système est basé sur le protocole Z-Wave ; il est composé de 43 dispositifs matériels et plusieurs couches et composantes logicielles.

Les résultats montrent que les analyses proposées nous ont permis d'identifier et de localiser un certain nombre d'anomalies, notamment fantômes et anti-causales.

Notre analyse, effectuée en novembre-décembre 2019, a confirmé la présence de tous les types d'anomalies définies, avec un pic de lecture ponctuelle d'un capteur hors gamme tous les deux jours. Dans la même période, nous avons trouvé des preuves de plusieurs problèmes anticausales et fantômes.

Cependant, le lien entra les pics d'activités et les anomalies reste faible et/ou à prouver. Notre approche est inspiré du N-version programming : utiliser trois experts pour identifier des pics d'activités. Cela nous a permis de prouver la présence de pics d'activités qui vont jusqu'à 50 fois pour heure.

Le système mis en place est non-supervisé et donc flexible. Cela le rend facile à appliquer dans d'autres contextes. Un premier essai fait pour expliquer le retard entre un mouvement détecté et une lumière activée avec les pics d'activités nous a donné une valeur de corrélation de seulement 30 %. Il est donc évident que les pics d'activités ont un impact sur le comportement du système et sur réseau Z-Wave, mais, la majorité des phénomènes restent sans explications.

De la même façon nous avons trouvé que la notion de voisinage et les matrices de voisinage ne sont pas suffisantes pour expliquer la stratégie de routage ou le nombre de passages intermédiaires entre un dispositif et le contrôleur. Les matrices de voisinage ne changent pas en temps normal. Cela arrive seulement si nous ajoutons ou retirons des noeuds. Donc, il est évident que le système utilise d'autres informations pour décider de quelle façon les messages sont envoyé.

7.2 Limitations de la solution proposée

Un autre grand problème est celui de la dénomination des appareils et de la rotation des noms des appareils. La chaîne ZWayVDev_zway_62-0-37 identifie l'interrupteur du salon.

L'utilisateur peut créer un alias LivingRoom, mais le système 1) utilisera toujours 62 et ZWayVDev_zway_62-0-37 comme ID de dispositif ; et 2) si par hasard le dispositif 62 est classé comme échoué, l'ID 62 peut être réaffecté à un dispositif différent. Après un certain temps, il est frustrant d'essayer de comprendre quel appareil se trouve dans quelle pièce : l'appareil (plus précisément l'identifiant de l'appareil) disparu de la salle à manger peut réapparaître dans le salon. Une des limitations de notre travail est que nous avons été obligé de travailler avec l'association pièce-dispositif vu que l'identifiant du dispositif n'est pas fiable.

De plus, nous avons travaillé avec un système réel, mais nous ne pouvons pas être certains que d'autres systèmes avec la même technologie (Z-Wave) ou une technologie différente (ZigBee) présentent les mêmes problématiques.

Une autre limite est lié à l'absence des détails sur la couche bas niveau aisément accessible pour le protocole Z-Wave . En effet, nous n'avons pas réussi à extraire les tables de routages ni à forcer certains parcours par défaut. Cette partie reste a explorer dans nos travaux futurs.

La création de l'oracle était forcement manuelle et subjective. Plusieurs personnes pourront interpréter les mêmes séries temporelles de manières différentes. De plus, pour des raisons de fiabilité et simplicité de l'interprétation, nous nous sommes concentrés sur la période d'hiver 2019. Le nombre de séries temporelles est, évidemment, limité. De plus, à partir du mois de janvier, la maison était vide donc les séries temporelles ne sont pas représentatives. Plus de données sont essentielles pour vérifier la précision des résultats.

Une dernière limitation est liée à la difficulté d'obtenir des données fiables et les associer à des causes réelles. Par exemple, nous avons observé que certains dispositifs après avoir atteint un bas niveau de batterie ou bien étant débranchés de l'alimentation, redémarrent avec la configuration par défaut. Malheureusement, cette configuration peut ne pas être souhaitée. Les capteurs multi-sensor ont redémarré avec un intervalle de mesure d'une heure, tandis qu'en réalité la configuration programmée était d'une mesure toutes les cinq minutes. Pour la température, une seule valeur par heure n'est pas considérée comme précise. Du coup, nous avons perdu plusieurs jours de mesures en étant convaincus que nous avions toujours la bonne configuration.

Néanmoins, la limitation majeure est dans la nature de notre travail préliminaire. Nous nous sommes concentrés sur la détection des anomalies : tout le travail pour améliorer et pallier les problèmes est à faire.

7.3 Améliorations futures

Les leçons pour les concepteurs et les développeurs sont nombreuses. Nous pouvons détecter leurs éventuelles défaillances, notamment les anomalies anti-causales et fantômes. Toutefois, ces fonctionnalités doivent être intégrées nativement dans les systèmes et les applications. L'utilisateur doit avoir la possibilité de les activer et les désactiver. Le système (et les applications) doivent produire un rapport quotidien (ou hebdomadaire) succinct et visuel de ce qui se passe, avec une identification claire des problèmes éventuels et des mesures prises pour y remédier. Par exemple, les anomalies ponctuelles sont faciles à détecter en utilisant des contraintes physiques. Il s'agit, tout simplement, de la bonne pratique des validations des entrées. Pourtant, il n'est pas encore mis en œuvre dans les plates-formes d'automatisation à domicile disponibles.

En continuant sur la même voie, les horaires et les scènes devraient générer automatiquement des rapports, mettant en évidence les actions et les anomalies potentielles. Un horaire (ou une scène) devrait générer automatiquement un observateur qui surveille le comportement de l'horaire (ou de la scène) donné et son effet sur le monde réel. Des diagrammes inspirés de la figure 6.7 seront utiles, ainsi que des graphiques similaires à la figure 6.2, pour résumer, rapidement, l'état de santé du système.

Pour finir, les systèmes et les applications qui s'exécutent sur ces appareils, ainsi que les microprogrammes de ces derniers, doivent être conçus et développés en tenant compte du modèle transactionnel. À l'heure actuelle, en cas de panne de courant, lorsque le système redémarre, il n'est pas garanti que son état et les appareils associés soient de retour comme prévu. Une coupure de courant est un problème majeur qui nécessite un certain temps pour déconnecter et retirer un appareil du système et ensuite réintégrer l'identification. Cependant, à ce stade, il est très probable que l'identifiant de l'appareil soit modifié.

comme correctif temporaire, des observateurs spécifiques (voir le modèle fail2ban) doivent être déployés pour les fonctions critiques. Nous disposons d'un premier ensemble de Linux daemons pour surveiller la santé globale du système.

RÉFÉRENCES

[1] M. U. Togbe, Y. Chabchoub, A. Boly et R. Chiky, "Etude comparative des méthodes de détection d'anomalies," *Revue des Nouvelles Technologies de l'Information*, févr. 2020. [En ligne]. Disponible : https://hal.archives-ouvertes.fr/hal-02874904

[2] K. DOUAIOUI, M. FRI, C. MABROUKKI et E. A. SEMMA, "The interaction between industry 4.0 and smart logistics : concepts and perspectives," dans *2018 International Colloquium on Logistics and Supply Chain Management (LOGISTIQUA)*, vol. 00212667984883, 2018, p. 128–132.

[3] C. Paetz, *Z-Wave Basics : Remote Control in Smart Homes*. North Charleston, SC, USA : CreateSpace Independent Publishing Platform, 2013.

[4] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld et M. Hoffmann, "Industry 4.0," *Business & information systems engineering*, vol. 6, n°. 4, p. 239–242, 2014.

[5] A. Cachada, J. Barbosa, P. Leitño, C. A. Gcraldcs, L. Deusdado, J. Costa, C. Teixeira, J. Teixeira, A. H. Moreira, P. M. Moreira *et al.*, "Maintenance 4.0 : Intelligent and predictive maintenance system architecture," dans *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1. IEEE, 2018, p. 139–146.

[6] J. Li, C. Xuan, B. Shao, H. Ji et C. Ren, "A new connected device-based failure mode and effects analysis model," dans *Proceedings of 2014 IEEE International Conference on Service Operations and Logistics, and Informatics*. IEEE, 2014, p. 1–5.

[7] Y. Zhang, N. Dragoni et J. Wang, "A framework and classification for fault detection approaches in wireless sensor networks with an energy efficiency perspective," *International Journal of Distributed Sensor Networks*, vol. 11, n°. 11, p. 678029, 2015.

[8] D. Stiawan, M. Y. Idris, R. F. Malik, S. Nurmaini et R. Budiarto, "Anomaly detection and monitoring in internet of things communication," dans *2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 2016, p. 1–4.

[9] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buonadonna, D. Gay *et al.*, "A macroscope in the redwoods," dans *Proceedings of the 3rd international conference on Embedded networked sensor systems*, 2005, p. 51–63.

[10] S. Agrawal et J. Agrawal, "Survey on anomaly detection using data mining techniques," *Procedia Computer Science*, vol. 60, p. 708 – 713, 2015, knowledge-Based and

Intelligent Information & Engineering Systems 19th Annual Conference, Singapore, September 2015 Proceedings. [En ligne]. Disponible : http://www.sciencedirect.com/science/article/pii/S1877050915023479

[11] E. Keogh, J. Lin et A. Fu, "Hot sax : Efficiently finding the most unusual time series subsequence," 12 2005, p. 8 pp.–.

[12] H. Cheng, P.-N. Tan, C. Potter et S. Klooster, *Detection and Characterization of Anomalies in Multivariate Time Series*, 04 2009, p. 413–424.

[13] H. Qiu, Y. Liu, N. A. Subrahmanya et W. Li, "Granger causality for time-series anomaly detection," dans *Proceedings of the 2012 IEEE 12th International Conference on Data Mining*, ser. ICDM '12. Washington, DC, USA : IEEE Computer Society, 2012, p. 1074–1079. [En ligne]. Disponible : https://doi.org/10.1109/ICDM.2012.73

[14] N. Laptev, S. Amizadeh et I. Flint, "Generic and scalable framework for automated time-series anomaly detection," dans *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '15. New York, NY, USA : ACM, 2015, p. 1939–1947. [En ligne]. Disponible : http://doi.acm.org/10.1145/2783258.2788611

[15] D. Kwon, H. Kim, J. Kim, S. Suh, I. Kim et K. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Computing*, vol. 22, 01 2019.

[16] L. Rabiner, A. Rosenberg et S. Levinson, "Considerations in dynamic time warping algorithms for discrete word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, n°. 6, p. 575–582, December 1978.

[17] H. Sakoe, "Two-level dp-matching–a dynamic programming-based pattern matching algorithm for connected word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, n°. 6, p. 588–595, December 1979.

[18] A. Guzman et A. Gupta, *IoT Penetration Testing Cookbook : Identify Vulnerabilities and Secure Your Smart Devices*. Packt Publishing, 2017.

[19] B. Russell et D. V. Duren, *Practical Internet of Things Security : Design a Security Framework for an Internet Connected Ecosystem, 2nd Edition*, 2e éd. Packt Publishing, 2018.

[20] A. blueGupta, *The IoT Hacker's Handbook A Practical Guide to Hacking the Internet of Things*. Apress, 2019.

[21] I. Unwala, Z. Taqvi et J. Lu, "Iot security : Zwave and thread," dans *2018 IEEE Green Technologies Conference (GreenTech)*, 2018, p. 176–182.

[22] L. D. Xu, W. He et S. Li, "Internet of things in industries : A survey," *IEEE Transactions on Industrial Informatics*, vol. 10, n°. 4, p. 2233–2243, 2014.

[23] H. Boyes, B. Hallaq, J. Cunningham et T. Watson, "The industrial internet of things (iiot) : An analysis framework," *Computers in Industry*, vol. 101, p. 1 – 12, 2018. [En ligne]. Disponible : http://www.sciencedirect.com/science/article/pii/S0166361517307285

[24] U. Ahmad, J. Chaudhary et A. Naz, "Survey on internet of things (iot) for different industry environments," *Annals of Emerging Technologies in Computing*, vol. 3, p. 28–43, 07 2019.

[25] P. P. Ray, "A survey of iot cloud platforms," *Future Computing and Informatics Journal*, vol. 1, n°. 1, p. 35 – 46, 2016. [En ligne]. Disponible : http://www.sciencedirect.com/science/article/pii/S2314728816300149

[26] M. Ammar, G. Russello et B. Crispo, "Internet of things : A survey on the security of iot frameworks," *Journal of Information Security and Applications*, vol. 38, p. 8 – 27, 2018. [En ligne]. Disponible : http://www.sciencedirect.com/science/article/pii/S2214212617302934

[27] S. Al-Sarawi, M. Anbar, K. Alieyan et M. Alzubaidi, "Internet of things (iot) communication protocols : Review," dans *2017 8th International Conference on Information Technology (ICIT)*, 2017, p. 685–690.

[28] Yu Liu, Kin-Fai Tong, Xiangdong Qiu, Ying Liu et Xuyang Ding, "Wireless mesh networks in iot networks," dans *2017 International Workshop on Electromagnetics : Applications and Student Innovation Competition*, 2017, p. 183–185.

[29] K. Georgiou, S. Xavier-de-Souza et K. Eder, "The iot energy challenge : A software perspective," *IEEE Embedded Systems Letters*, vol. 10, n°. 3, p. 53–56, 2018.

[30] N. Javaid, S. Cheema, M. Akbar, N. Alrajeh, M. S. Alabed et N. Guizani, "Balanced energy consumption based adaptive routing for iot enabling underwater wsns," *IEEE Access*, vol. 5, p. 10 040–10 051, 2017.

[31] Q. Wu, W. Chen, D. W. K. Ng et R. Schober, "Spectral and energy-efficient wireless powered iot networks : Noma or tdma ?" *IEEE Transactions on Vehicular Technology*, vol. 67, n°. 7, p. 6663–6667, 2018.

[32] E. Anthi, L. Williams, M. Słowińska, G. Theodorakopoulos et P. Burnap, "A supervised intrusion detection system for smart home iot devices," *IEEE Internet of Things Journal*, vol. 6, n°. 5, p. 9042–9053, Oct 2019.

[33] M. Jackson et P. Zave, "Distributed feature composition : a virtual architecture for telecommunications services," *IEEE Transactions on Software Engineering*, vol. 24, n°. 10, p. 831–847, Oct 1998.

[34] M. H. Zibaeenejad, C. Zhang et J. M. Atlee, “Continuous variable-specific resolutions of feature interactions,” dans *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, ser. ESEC/FSE 2017. New York, NY, USA : Association for Computing Machinery, 2017, p. 408–418. [En ligne]. Disponible : https://doi.org/10.1145/3106237.3106302

[35] R. B. Abdessalem, A. Panichella, S. Nejati, L. C. Briand et T. Stifter, “Testing autonomous cars for feature interaction failures using many-objective search,” dans *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, ser. ASE 2018. New York, NY, USA : Association for Computing Machinery, 2018, p. 143–154. [En ligne]. Disponible : https://doi.org/10.1145/3238147.3238192

[36] M. M. Breunig, H.-P. Kriegel, R. T. Ng et J. Sander, “Lof : Identifying density-based local outliers,” dans *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD ’00. New York, NY, USA : Association for Computing Machinery, 2000, p. 93–104. [En ligne]. Disponible : https://doi.org/10.1145/342009.335388

[37] M. Crawley, *The R book*. Wiley, 2013.

[38] P. K. Sen, “Estimates of the regression coefficient based on kendall’s tau,” *Journal of the American Statistical Association*, vol. 63, p. 1379–1389, 01 1968.

[39] J. Lanzante, “Resistant, robust and non-parametric techniques for the analysis of climate data : Theory and examples, including applications to historical radiosonde station data,” *International Journal of Climatology*, vol. 16, p. 1197–1226, 01 1996.

[40] D. R. Cox et A. Stuart, “Some quick sign tests for trend in location and dispersion ,” *Biometrika*, vol. 42, n°. 1-2, p. 80–95, 06 1955. [En ligne]. Disponible : https://doi.org/10.1093/biomet/42.1-2.80

[41] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell et A. Wesslén, *Experimentation in Software Engineering - An Introduction*. Kluwer Academic Publishers, 2000.

[42] V. Basili, G. Caldiera et D. H. Rombach, *The Goal Question Metric Paradigm Encyclopedia of Software Engineering*. John Wiley and Sons, 1994.

[43] L. Kaufman et P. Rousseeew, *Finding groups in data : an introduction to cluster analysis*. Wiley - NY : Wiley-Inter Science, 1990.

[44] A. V. Dian Sano, T. Daud Imanuel, M. Intanadias Calista, H. Nindito et A. Raharto Condrobimo, “The application of agnes algorithm to optimize knowledge base for tourism chatbot,” dans *2018 International Conference on Information Management and Technology (ICIMTech)*, 2018, p. 65–68.