Université Abou Bekr Belkaid
Tlemcen Algérie
جامعة أبي بكر بلقايد
تلمسان الجزائر

République Algérienne Démocratique et Populaire

Université Abou Bakr Belkaid– Tlemcen

Faculté desSciences

Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Informatique

Option: Réseaux et Systèmes Distribués (R.S.D)

Thème

La détection des anomalies dans la consommation d'électricité en utilisant des méthodes de détection des outliers

Réalisé par :

KARZAZI Hayat

LARBAOUI Maghnia

Présenté le 06 Juillet 2019 devant le jury composé de :

- *Mr BENMAMMAR Badr* — *Président*
- *Mr LEHSAINI Mohamed* — *Encadrant*
- *Mr BENMOUNA Youcef* — *Examinateur*
- *Mr KADDOUR Sidi Mohammed* — *Co-encadrant*

Année universitaire:2018-2019

Remerciements

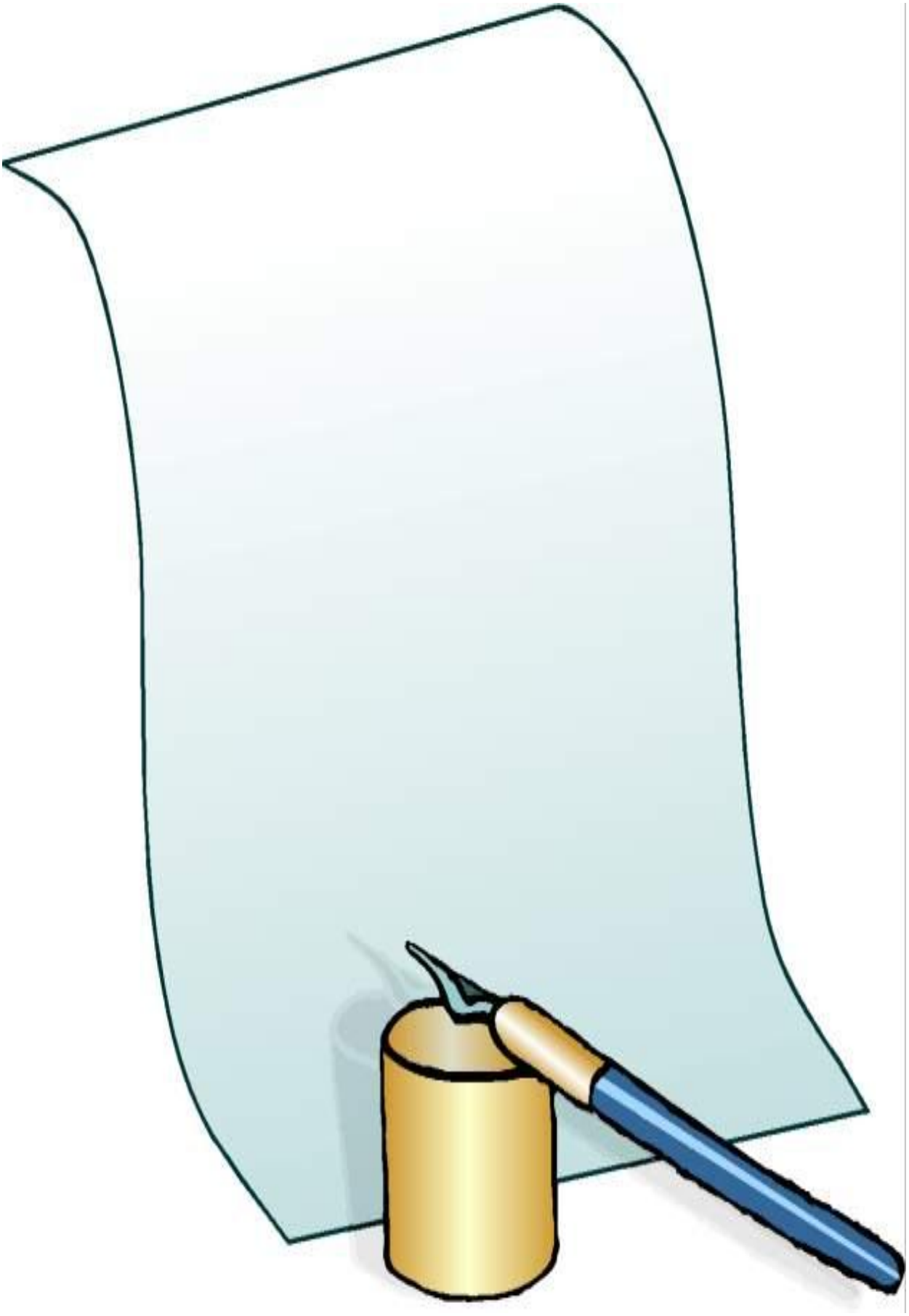
Tout d'abord, je tiens à remercier le bon Dieu le tout Puissant de m'avoir donné la force et le courage de mener à bien ce modeste travail

Au terme de ce travail, nous tenons à exprimer notre profonde gratitude à notre professeur et encadreur M^er^ LEHSAINI Mohamed Directeur de Laboratoire de Recherche STIC pour son suivi et pour son énorme soutien qu'il n'a cessé de nous orienter tout au long de la période du projet.

Nous tenons à remercier également notre encadrant *M^er^ KADDOUR Sidi Mohamed* doctorant à l'université Abou Baker Belkaid pour le temps qu'il a consacré et pour les précieuses informations qu'il nous a accordé avec intérêt et compréhension.

Nous adressons aussi nos vifs remerciements aux membres des jurys pour avoir bien voulu examiner et juger ce travail.

Nous tenons aussi à remercier tous ce qui nous a aidé de près ou de loin pour réaliser ce travail.



SOMMAIRE

Liste des Figures

Liste des Tableaux

Introduction générale

Introduction générale

Dans le marché d'électricité, la connaissance des consommateurs d'électricité fournit une compréhension de leur comportement de consommation, qui est récemment devenu important dans l'industrie électrique. Avec cette connaissance, les fournisseurs d'électricité sont capables de développer une nouvelle stratégie commerciale et d'offrir des services basés sur la demande des clients.

La méthode la plus efficace actuellement pour réduire les pertes commerciales est d'utiliser des compteurs électroniques intelligents. Ces compteurs peuvent aviser les consommateurs en tout moment si sa consommation dépasse un certain seuil.

Les valeurs aberrantes étaient considérées comme des données statistiques bruyantes, et s'est avéré être un problème important qui fait l'objet de recherches dans divers domaines de recherche et d'applications. Beaucoup techniques de détection des valeurs aberrantes ont été développées spécifiquement pour certains domaines d'application, alors que certaines techniques sont plus génériques. Certains domaines d'application font l'objet de recherches très strictes.

Le but de ce projet est d'utiliser une méthode de détection des anomalies dans les séries temporelles pour identifier des défaillances électriques et l'énergie gaspillée dans les bâtiments résidentielles

Ce mémoire est structuré en quatre chapitres comme suit :

- Le premier chapitre présente des notions générales sur le smart grid (le réseau intelligent) et le système de gestion de l'énergie domestique,
- Le deuxième chapitre est une brève description des méthodes de Machine Learnig (ML) les plus répandues dans la littérature qui permettent la détection des anomalies dans les systèmes.
- Le troisième chapitre présente les outils logiciels tels que la plateforme analytique KNIME, les bibliothèques de Python utilisées dans l'apprentissage automatique et la base de données introduite pour la détection des anomalies ainsi que les méthodes implémentées (One-Class SVM, K-Means, Isolation Forest).
- Le quatrième chapitre présente l'implémentation du modèle développé en utilisant la plateforme par le système KNIME et trois méthodes issues de l'apprentissage non supervisé pour la détection des anomalies.

Cette partie importante illustra le projet général et les méthodologies de détection. Les trois grandes étapes qui seront impliquées dans le développement du système de détection des anomalies, notamment : la lecture des données, le prétraitement des données, l'analyse des données par les trois méthodes et enfin la visualisation et la comparaison des résultats pour déterminer la meilleure méthode.

Enfin, on conclut ce mémoire par un récapitulatif sur le travail réalisé dans le cadre de notre projet de fin d'études.

Chapitre I

Les réseaux électriques intelligents

(Smart Grids)

Chapitre I
Les réseaux électriques intelligents (Smart Grids)

I.1 Introduction

Le réseau électrique traditionnel fonctionne de manière centralisée et de manière unidirectionnelle, du producteur vers le consommateur. L'équilibre du système électrique est obtenu en tenant compte de l'offre d'énergie en fonction de la demande tout en minimisant le coût de production de l'électricité. Par ailleurs, le développement des énergies renouvelables et l'évolution des usages de l'électricité (climatisation, chauffage, voiture électrique) exigent la modernisation de ce réseau pour répondre aux besoins des usagers en termes d'énergie.

Ces nouveaux usages contraignent le pilotage des réseaux électriques en raison de :

- La consommation d'électricité connaît de fortes variations suivant les conditions climatiques. Par exemple, en Algérie, la consommation d'énergie est plus importante en été qu'en hiver à cause de l'utilisation de la climatisation qui consomme plus d'un usage ordinaire. Cette consommation fait l'objet de pointes et de creux journaliers en particulier quand la température atteint des degrés élevés.
- Les moyens de production d'électricité sont de plus en plus variables, du fait de l'intermittence de leurs sources renouvelables.
- Le développement de la production décentralisée conduit à multiplier de manière très importante les sites de production, et à injecter de l'énergie sur des réseaux de distribution conçus pour l'acheminer et non la collecter c'est-à-dire la production de l'énergie suivant la consommation.

Ces contraintes imposent de revoir les règles classiques d'exploitation des réseaux et exigent qu'on les adapte en termes de conduite des réseaux électriques. Le réseau électrique actuel date de plusieurs décennies quand les besoins de la vie des usagers étaient limités cependant avec les besoins grandissants des usagers il est devenu nécessaire de le moderniser et penser à mettre en place des réseaux intelligents "Smart Grids".

I.2 Les Smart Grids

I.2.1 Définition

Le terme "Smart grids" désigne le réseau de distribution d'électricité où le mot "Smart" met l'accent sur "l'intelligence" pour le caractériser des réseaux électriques classiques. La notion de "Smart grid" peut ainsi se traduire par "réseau électrique intelligent".

D'après la commission européenne[1], les "smart grids" sont des réseaux électriques intelligents capables d'intégrer efficacement les comportements et actions de tous les utilisateurs qui y sont raccordés – producteurs, consommateurs et utilisateurs à la fois producteurs et consommateurs – afin de constituer un système rentable et durable, présentant des pertes faibles et un niveau élevé de qualité et de sécurité d'approvisionnement [1].

I.2.2 Caractéristiques des réseaux intelligents

Les réseaux électriques intelligents intègrent des fonctionnalités issues des technologies de l'information et de la communication qui permettent à ces réseaux d'établir des communications entre les différents points des réseaux en tenant compte des actions des différents acteurs du système électrique en particuliers les consommateurs. L'objectif principal des réseaux électriques intelligents est d'assurer l'équilibre entre l'offre et la demande à tout instant et d'optimiser le fonctionnement des réseaux en fonction de cette demande.

Les réseaux intelligents peuvent être définis selon quatre caractéristiques :

- Flexibilité : ils permettent de gérer avec une grande précision l'équilibre entre production et consommation.
- Fiabilité : ils améliorent l'efficacité et la sécurité des réseaux tout en évitant une éventuelle défaillance entre l'offre et la demande à des instants bien précis.
- Accessibilité : ils favorisent l'intégration des sources d'énergies renouvelables variées sur l'ensemble du réseau tout en optimisant les coûts de production.
- Economie : ils apportent, grâce à une meilleure gestion du système, des économies d'énergie et une diminution des coûts (à la production comme à la consommation).

Le tableau 1 présente une comparaison entre les réseaux électriques classiques et les réseaux électriques intelligents.

[1] Communication "Smart grids de l'innovation au déploiement", Commission européenne, Avril 2011

Tableau 1: Comparaison entre les réseaux électriques traditionnels et intelligents [2]

Réseaux électriques classique	Réseaux électriques intelligents
Analogique	Numérique
Unidirectionnel	Bidirectionnel
Production centralisée	Production décentralisée
Communicant sur une partie des réseaux	Communicant sur l'ensemble des réseaux
Gestion de l'équilibre du système électrique par l'offre/ production	Gestion de l'équilibre du système électrique par la demande/consommation

I.2.3 Fonctionnement

Un réseau intelligent associe l'infrastructure électrique à la technologie de l'information et de la communication qui permet d'analyser et de transmettre l'information reçue. Cette technologie de communication est impliquée à tous les niveaux du réseau : production, transport, distribution et consommation [3].

Les réseaux électriques intelligents permettent :

- **Le contrôle des flux en temps réel** : Ce contrôle est réalisé par le biais des capteurs installés sur l'ensemble du réseau. Ces derniers indiquent instantanément les flux électriques et les niveaux de consommation. Les opérateurs du réseau peuvent alors réorienter les flux énergétiques en fonction de la demande et envoyer des signaux de prix aux particuliers pour adapter leur consommation selon leurs choix ou d'une manière automatique.
- **L'interopérabilité des réseaux :** l'ensemble du réseau électrique comprend le réseau de transport et le réseau de distribution. Le premier relie les sites de production d'électricité aux zones de consommation : ce sont les grands axes qui quadrillent le territoire. Le réseau de distribution s'apparente aux axes secondaires. Il achemine l'électricité jusqu'aux consommateurs finaux. Par l'échange instantané d'informations, les smart grids favorise une interopérabilité entre les gestionnaires du réseau de transport et ceux du réseau de distribution pour permettre à ces deux systèmes de se cohabiter.
- **L'intégration des énergies renouvelables au réseau :** les réseaux électriques intelligents reposent sur un système d'information qui permet de prévoir à court et à long terme le niveau de production et de consommation. Les énergies renouvelables qui fonctionnent souvent de

façon peu prévisible peuvent ainsi être mieux gérées pour satisfaire les besoins des consommateurs à moindre coût tout en favorisant les sources d'énergies moins polluantes.

- **Une gestion plus responsable des consommations individuelles :** les compteurs communicants sont les premières versions d'application du réseau intelligent. Installés chez les consommateurs, ils fournissent des informations sur les prix, les heures de pointe de consommation, la qualité et le niveau de consommation d'électricité du foyer. Les consommateurs peuvent alors réguler eux-mêmes leur consommation au cours de la journée. De leur côté, les opérateurs du réseau peuvent détecter plus vite les pannes en fonction du comportement des usagers en termes de consommation de l'électricité.

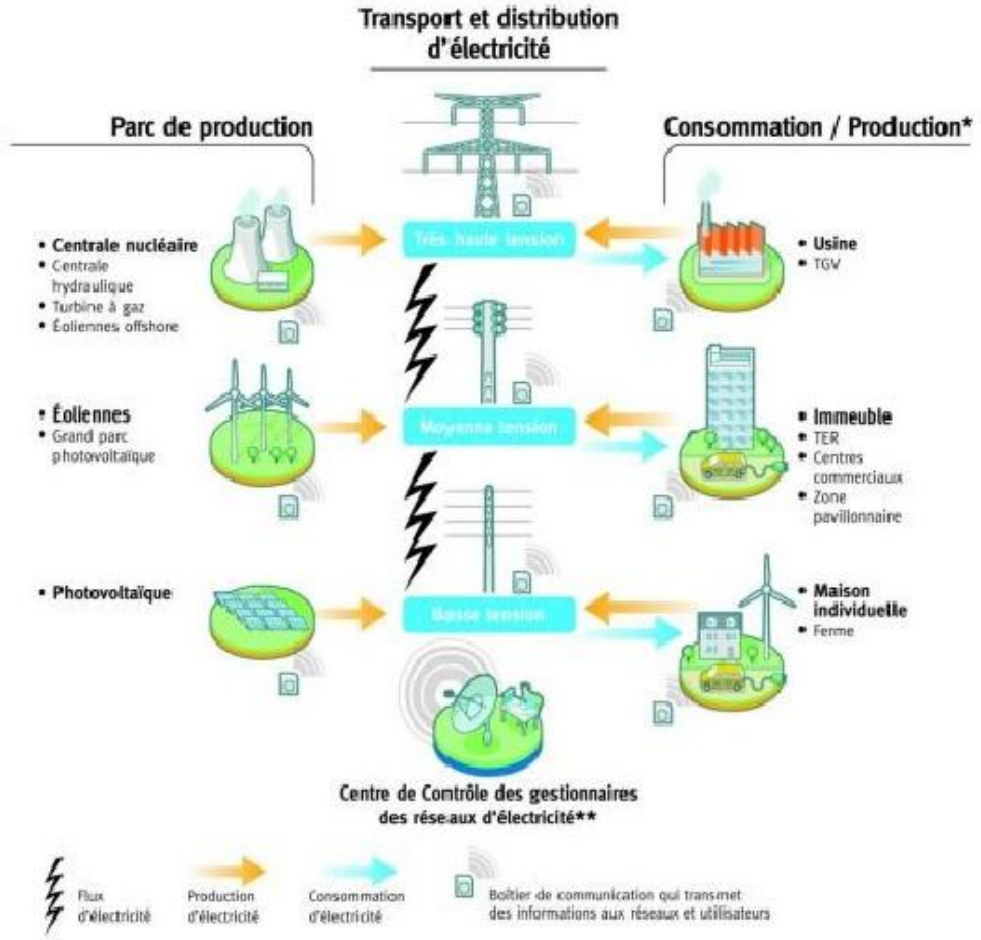


Figure 1 : Fonctionnement des Smart Grids [3]

1.3 Intérêts des réseaux électriques intelligents

L'électricité ne peut être stockée facilement et économiquement en grandes quantités. A cet effet, les technologies utilisées dans les réseaux électriques intelligents cherchent à ajuster en temps réel la

production et la distribution (offre et demande) de l'électricité en hiérarchisant les besoins de consommation (quantité et localisation) selon leur urgence afin de [4] :

- optimiser le rendement des centrales et/ou petites unités de production de l'électricité.
- éviter d'avoir à construire régulièrement de nouvelles lignes de production.
- minimiser les pertes en ligne.
- optimiser l'insertion de la production décentralisée et diminuer ou éliminer les problèmes induits par l'intermittence de certaines sources (énergies solaires, éolien, marémotrice et moindrement hydroélectricité) tant en optant aux sources d'énergie moins polluantes.

I.4 Caractéristiques des réseaux de capteurs sans fil

Les opérateurs de l'électricité, les fournisseurs d'énergie (y compris renouvelable), les transporteurs d'énergie et les consommateurs sont confrontés à de nombreux défis en particulier dans quatre principaux domaines où ils souhaiteraient voir des améliorations [5] :

- **Efficacité** : Les fournisseurs de l'énergie souhaitent produire l'énergie à moindre coût et transporter le maximum d'énergie possible, avec le minimum de pertes et de goulets d'étranglement. Ils doivent connaitre les points de congestion dus au type et à la capacité des câbles, ou encore à l'architecture du réseau. Ils souhaitent connaître aussi la quantité d'énergie consommée par leurs clients et à quel moment, région par région, mais aussi alimenter leur réseau à partir de sources d'énergie renouvelables à tous les niveaux, en améliorant la gestion de la charge, éventuellement grâce à une régulation dynamique en fonction de la consommation instantanée par client et/ou par région.

- **Fiabilité et sécurité** : L'électricité doit être distribuée de manière fiable et d'une manière continue, c'est-à-dire sans interruptions, ni variations de fréquence ou fluctuations de qualité (pics ou chutes de tension). Les opérateurs souhaitent réduire le nombre des pannes et des coupures, ainsi que leur durée. La sécurité doit permettre de détecter rapidement toute surchauffe de câbles qui pourra causer une éventuelle panne matérielle. Le vol d'énergie peut également constituer un problème crucial pour les opérateurs d'électricité et les interventions au niveau des postes doivent être mieux maîtrisées. Dans la mesure du possible, les situations anormales, de type surchauffe ou court-circuit, doivent pouvoir être détectées et corrigées de manière automatique et dans les meilleurs délais.

- **Souplesse** : Lorsque la saturation du réseau entraîne des problèmes de distribution, les fournisseurs d'énergie souhaitent pouvoir rerouter la distribution via d'autres lignes, partager ou

importer de l'électricité des autres points de distribution pour faire face aux variations de la demande. Cela exige un suivi en continu et une supervision du réseau pour prédire son comportement. Les données vitales des utilisateurs doivent être accessibles via une infrastructure de compteurs intelligents et de fibres optiques à haut débit pour les aviser en cas d'un mode de consommation inadéquat à la normale.

- **Topologie dynamique** : Les opérateurs et leurs clients sont sensibles aux questions environnementales. Ils doivent obtenir la garantie que les lignes électriques enfouies ou aériennes sont sécurisées et engendrent un minimum de pertes, d'émissions de CO2 et d'interférences électromagnétiques. Le réseau plus intelligent doit permettre l'interaction avec de multiples types de véhicules électriques, ainsi que la production et la consommation d'énergie propre au niveau du client.

I.5 Maison Intelligente

Pour s'approvisionner en énergie à travers les réseaux électriques intelligents et pouvoir communiquer avec les îlots "Smart City" tout en respectant le droit à la confidentialité des occupants, le bâtiment ou la maison doit être obligatoirement communiquant pour mériter d'être "Smart Home". Une première application de bâtiment intelligent est apparue aux Etats-Unis dans les années 1970 sous le nom de "Building Energy Management System (BEMS)". L'idée de "Smart Home" s'est consolidée dans les années 1980, avec les développements des technologies de l'information et de la communication.

D'après F-X.Jeuland, une maison intelligente est une maison qui dispose de fonctionnalités susceptibles de simplifier la vie de ses occupants au quotidien, de réaliser des économies d'énergie et d'apporter un certain niveau de confort et de sécurité. Elle est bien préparée aux évolutions futures par la nature même de ses infrastructures de câblage et par son ouverture au monde numérique.

I.6 Système de gestion d'énergie (EMS : Energy Management System)

Un système de gestion d'énergie, appelé EMS est un système d'outils informatiques utilisés par les opérateurs d'électricité pour surveiller, contrôler et optimiser les performances de la génération ou système de transmission .

L'appellation "système de gestion de l'énergie" peut également désigner un système informatique conçu spécifiquement pour le contrôle et la surveillance automatisés des installations électromécaniques d'un bâtiment générant une consommation d'énergie importante, telles que les installations de chauffage, de climatisation et d'éclairage. Le champ d'application peut aller d'un seul

bâtiment à un groupe de bâtiments tels que des campus universitaires, des immeubles de bureaux ou des usines. La plupart de ces systèmes de gestion de l'énergie fournissent également des installations pour la lecture des compteurs d'électricité, de gaz et d'eau. Les données obtenues à partir de celles-ci peuvent ensuite être utilisées pour effectuer fréquemment des routines d'autodiagnostic et d'optimisation et pour produire une analyse des tendances et des prévisions de consommation annuelle chez les individus ou globale.

I.7 Système de gestion de l'énergie domestique (HEMS)

I.7.1 Gestion Technique de bâtiment (GTB)

La gestion technique de bâtiment (GTB) est un système automatisé permettant de superviser les différents équipements électriques et mécaniques d'un bâtiment, comme la ventilation, la température, l'éclairage, l'alimentation électrique, les systèmes de sécurité ou encore les systèmes anti-incendie. Ce mode de gestion peut ainsi commander les autorisations d'accès aux bâtiments, remonter les alarmes déclenchées en cas d'anomalies et permet de faire le suivi des consommations d'énergie et d'eau chez les individus ou d'une région toute entière.

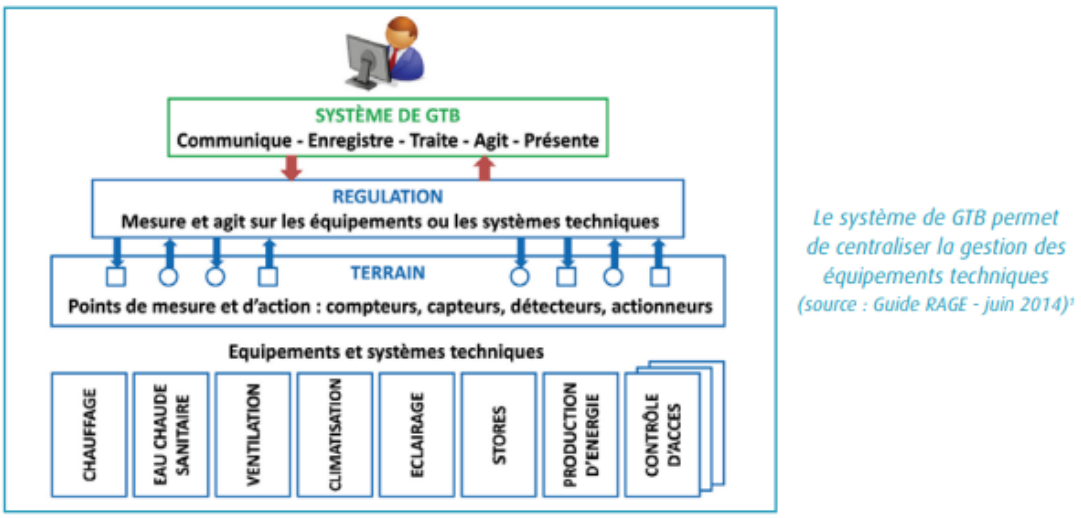


Figure 2: Système de gestion de l'énergie domestique [6]

I.7.2 Les objectifs principaux d'une GTB

Les objectifs principaux d'une GTB sont :

- Assurer la sécurité du bâtiment et de ses occupants ;
- Gérer le fonctionnement des installations techniques ;
- Maîtriser les consommations de flux.

I.7.3 Le principe de fonctionnement de la GTB

La GTB assure l'automatisation des bâtiments en pilotant l'ensemble ou une partie des équipements. L'interface entre le système et l'utilisateur est assuré par un logiciel de supervision qui permet de contrôler tous les équipements se trouvant dans un bâtiment. Il permet généralement une gestion et un suivi à distance via une connexion internet. En outre, dans ces bâtiments, des automates pilotent les installations en utilisant des actionneurs en fonction des informations transmises par les capteurs. Les fonctions traditionnellement commandées par la GTB sont le chauffage (production et régulation), la production d'eau chaude, la ventilation, l'éclairage, les stores, le contrôle des accès, etc… [6].

I.8 Conclusion

Dans ce chapitre introductif, nous avons présenté les réseaux électriques intelligents, leurs caractéristiques et les enjeux entre offre et demande dans ce type de réseaux. Nous avons aussi présenté pourquoi nous faisons appel aux réseaux électriques dans les bâtiments d'aujourd'hui et comment remédier aux anomalies causées par les réseaux électriques traditionnels.

Dans le chapitre suivant, nous décrivons certaines méthodes intelligentes permettant d'une part de détecter les anomalies des équipements au sein de l'habitat et d'autre part de prédire la consommation prévisionnelle en électricité dans un bâtiment ou une région.

Chapitre II
Apprentissage automatique sur les séries temporelles

Chapitre II
Apprentissage automatique sur les séries temporelles

II.1 Introduction

Machine Learning est un domaine captivant qui a envahi plusieurs disciplines comme les statistiques, l'optimisation, l'algorithmique ou le traitement du signal. Ce domaine permet de nous faire apprendre à résoudre un problème de manière automatique dans le cadre du traitement de la donnée en construisant un modèle obtenu directement à partir d'exemples. Machine Learning a été utilisé depuis des décennies dans la reconnaissance automatique de caractères ou les filtres anti-spam. Il sert aussi à protéger contre la fraude bancaire, identifier les visages dans le viseur de notre appareil photo, ou traduire automatiquement des textes d'une langue vers une autre en tenant compte de l'aspect sémantique et la spécificité de la langue. Dans les années à venir, le Machine Learning nous permettra vraisemblablement d'améliorer la sécurité routière grâce aux véhicules autonomes et la commande des équipements au sein de l'habitat à distance.

L'apprentissage automatique est une discipline qui fait partie de l'intelligence artificielle (IA). Cette discipline impliquent des outils et des concepts des statistiques, et fait partie aussi de la science des données. Pour traiter un problème de classification, un algorithme d'apprentissage observe des données étiquetées et a comme objectif d'apprendre une fonction qui sera en mesure de classifier automatiquement les données qui lui seront présentées dans le futur et de minimiser l'erreur pour classifier ces données.

L'apprentissage automatique consiste à utiliser des ordinateurs pour optimiser un modèle de traitement de l'information selon certains critères de performance à partir d'observations, que ce soit des données, des exemples ou des expériences passées [7]. Par ailleurs, lorsque l'on connait le bon modèle de traitement à utiliser, et qu'on n'pas besoin de faire de l'apprentissage, il peut être utile lorsque :

- On n'a pas d'expertise sur le problème qu'on compte le traiter (ex. robot navigant sur une autre planète),

- On a une expertise, mais on ne sait pas comment l'expliquer (ex. reconnaissance de visages ou de formes),
- Les solutions aux problèmes changent dans le temps (exemple le routage de paquets dans un réseau dont la topologie est dynamique),
- Les solutions doivent êtres personnalisées (exemple la biométrie rétinienne).

II.2 Les séries temporelles

Les séries temporelles sont des données numériques qui évoluent dans le temps, elles constituent une part importante des données produites et disponibles sur Internet dans de différents domaines tels que le comportement des utilisateurs du web, le comportement des occupants d'un habitat,, ou des données issues de réseaux de capteurs en vue d'appréhender et de contrôler certains phénomènes.

Les données temporelles sont également impliquées dans des applications plus classiques telles que l'analyse de signaux décrivant des paramètres physiologiques tels que IRM[2], EEG[3], et ECG[4], les profils d'expression de gènes, et les courbes de charge de consommation d'énergie. Il y a un bon moment que la communauté scientifique s'est penchée sur le traitement des séries temporelles, la disponibilité de grandes quantités de données est relativement récente et de nouveaux défis ont vu le jour dans ce contexte. Par ailleurs, l'objectif principal de l'analyse d'une série temporelle est la prévision de ses futures réalisations en se basant sur ses valeurs passées dans différents contextes.

Les principaux axes d'études autour des séries temporelles qui ont été proposés dans la littérature sont les suivants :

- **La prédiction :** étant donnée une série temporelle $X = \{x_1, x_2, \dots., x_t\}$ contenant t points, il s'agit de prédire la ou les valeurs suivantes : x_{t+1}, x_{t+2}, ... c'est-à-dire dans le futur.
- **La classification** : étant donnée une série temporelles X, il s'agit de l'affecter à une des (deux ou plus) classes prédéfinies.
- **La complétion :** étant donnée une série temporelle $X = \{x_1, x_2, \dots., x_t\}$ contenant t points et un masque m_i tel que m_i=1 si la valeur de x_i est connue et $m_i = 0$, sinon, il s'agit d'inférer la ou les valeurs manquantes, c'est-à-dire les valeurs pour lesquelle $m_i = 0$.
- **L'indexation** : étant donnée une série temporelle X ainsi qu'une mesure de similarité (ou dissimilarité) notée $D = (X, X')$ telle que $D = (X, X')$ est grand si les séries X et X' sont

[2] IRM : Radiologie et Imagerie Médicale
[3] EEG : ElectroEncéphaloGramme
[4] ECG : ÉlectroCardioGraphie

similaires et petit sinon, il s'agit de trouver la ou les séries temporelles les plus similaires dans une base de données donnée.

- **La segmentation** : étant donnée une série temporelle $X = \{x_1, x_2, \dots., x_t\}$ avec $\forall i, x_i \in \mathbb{R}$, il s'agit de trouver une approximation $\hat{X} = \{k_1, k_2, \dots., k_K\}$ avec $\forall i, k_i \in \mathbb{R}$ et $K \ll T$ et où $\hat{X}$ est une bonne approximation de X.
- **Le partitionnement :** il s'agit de regrouper des séries temporelles d'une base de données donnée en plusieurs partitions différentes selon une mesure de similarité (ou dissimilarité) notée $D = (A, B)$ telle que $D = (X, X')$ est grand si les séries X et X' sont similaires et petit sinon.
- **La détection d'anomalies :** étant donnée une série temporelle X que l'on considère comme étant "normale", déterminer quelles séries au sein d'une base de données contiennent une "anomalie".

Les séries temporelles sont des données ordonnées dans le temps et cet ordonnancement a une signification que l'on ne peut ignorer et représente une part importante des problèmes abordés en apprentissage automatique. Ainsi, les méthodes de fouille de données classiques ne sont pas efficaces si on les applique aux séries temporelles mais les méthodes qui respectent la temporalité de ce type de données sont considérées comme des méthodes prometteuses [8].

II.3 Types de Machine Learning

Le Machine Learning est un champ assez vaste, et nous dressons dans cette section une liste des plus grandes classes de problèmes auxquels il s'intéresse.

II.3.1 Apprentissage supervisé

L'apprentissage supervisé est considéré comme le type de problèmes de machine learning le plus facile à appréhender. Son but est d'apprendre à faire des prédictions, à partir d'une liste d'exemples étiquetés, c'est-à-dire accompagnés de la valeur à prédire. Les étiquettes servent de "professeur" et supervisent l'apprentissage de l'algorithme.

L'apprentissage est dit supervisé lorsque les données qui entrent dans le processus sont déjà catégorisées et que les algorithmes doivent s'en servir pour prédire un résultat en vue de pouvoir le faire plus tard lorsque les données ne seront plus catégorisées. On peut par exemple donner au système une liste des profils des clients contenant des habitudes d'achat, et expliquer à l'algorithme lesquels sont des clients habituels et lesquels sont des clients occasionnels. Une fois l'apprentissage terminé, l'algorithme devra pouvoir déterminer tout seul à partir d'un profil client à quelle catégorie celui-ci pourra appartenir.

La majorité des apprentissages automatiques utilisent un apprentissage supervisé.

L'apprentissage supervisé consiste en des variables d'entrée (x) et une variable de sortie (Y). Nous utilisons par la suite un algorithme pour apprendre la fonction de mapping de l'entrée à la sortie: $Y = f(X)$. Le but est d'appréhender si bien la fonction de mapping que, lorsque nous avons de nouvelles données d'entrée (x), nous pouvons prédire les variables de sortie (Y) pour ces données. Ce mode raisonnement est appelé l'apprentissage supervisé, car le processus d'un algorithme tirant parti du jeu de données d'apprentissage peut être considéré comme un enseignant supervisant le processus d'apprentissage. Nous connaissons les réponses correctes, l'algorithme effectue des prédictions itératives sur les données d'apprentissage et est corrigé par l'enseignant. L'apprentissage s'arrête lorsque l'algorithme atteint un niveau de performance acceptable [9].

Un algorithme d'apprentissage supervisé a comme objectif de généraliser pour des entrées inconnues ce qu'il a pu "apprendre" grâce aux données déjà traitées par des experts, ceci de façon "raisonnable" [10].

L'apprentissage supervisé est généralement effectué dans le contexte de la classification et de la régression.

II.3.2 Classification

Un problème de classification survient lorsque la variable de sortie est une catégorie, telle que "rouge", "bleu" ou "maladie" et "pas de maladie". Nous donnons dans ce qui suit quelques exemples de classification :

- Dans le secteur bancaire pour la détection de la fraude par carte de crédit (fraude, pas fraude).
- Détection de courrier électronique indésirable (spam, pas spam).
- Dans le domaine du marketing utilisé pour l'analyse du sentiment de texte (heureux, pas heureux), (gentil, pas gentil).
- En médecine, pour prédire si un patient a une maladie particulière ou non surtout quand il y a une épidémie dans une région.

La classification pourra prendre les formes suivantes :

a) Classification binaire

Nous utilisons des données étiquetées pour prédire à quelle *classe* un objet appartient, la classification binaire, permet de distinguer si un objet appartient ou non à une classe. Par exemple, dire si dans une image figure un arbre ou non. Dans ce qui suit, nous présentons quelques exemples de problèmes de classification binaire :

- Identifier si un email est un spam ou non.
- Identifier si une image contient ou non un objet bien particulier.

b) Classification multi-classe

En apprentissage multi-classe, nous considérons les problèmes de classification où l'espace d'entrée $\mathcal{X}$ est un ensemble quelconque, et où l'espace de sortie est un ensemble discret $y = \{1, 2, \dots, C\}$ où C est le nombre de classes. Nous présentons dans ce qui suit quelques exemples de classification multi-classe :

- Identifier en quelle langue un texte est écrit (il existe plusieurs langues)
- Identifier l'expression d'un visage parmi une liste prédéfinie de possibilités (colère, tristesse, joie, etc.).
- Identifier à quelle espèce appartient une plante.

II.3.3 Régression

Un problème de régression se pose lorsque la variable de sortie est une valeur réelle, telle que "le coût du dollar" ou "le poids" [11]. Exemples :

- Prédire le prix de l'immobilier dans une ville
- Prédire le cours de bourse
- Prédire le nombre de clics sur un lien.
- Prédire le nombre d'utilisateurs et utilisatrices d'un service en ligne à un moment donné.

Certains types de problèmes fondés sur la classification et la régression incluent la prévision et la prévision de séries temporelles, respectivement.

Nous présentons dans ce qui suit quelques méthodes d'apprentissage :

- Arbres de décision
- Méthode des k plus proches voisins,
- Machine à vecteurs de support (SVM),
- Naive Bayes
- Les réseaux de neurones
- Régression linéaire
- Régression vectorielle de support (SVR)
- Arbres de régression

II.3.4 Apprentissage semi-supervisé

L'apprentissage semi-supervisé consiste à apprendre des étiquettes à partir d'un jeu de données partiellement étiqueté. Un bon exemple est une archive de photos où seules certaines images sont étiquetées (par exemple, chien, chat, personne) et la majorité sont sans étiquette.

De nombreux problèmes d'apprentissage machine du monde réel sont sous-jacents à ce mode d'apprentissage. Ces problèmes se situent entre l'apprentissage supervisé et l'apprentissage non supervisé.

II.3.5 Apprentissage par renforcement

L'apprentissage par renforcement fait référence à une classe de problèmes d'apprentissage dont l'objectif est d'apprendre, à partir d'expériences, ce qu'il convient de faire en différentes situations, de façon à optimiser une récompense quantitative au cours du temps.

Un paradigme classique pour présenter les problèmes d'apprentissage par renforcement consiste à considérer un agent autonome, plongé au sein d'un environnement, et qui doit prendre des décisions en fonction de son état courant. En retour, l'environnement procure à l'agent une récompense, qui peut être positive ou négative. L'agent cherche, au travers d'expériences itérées, un comportement décisionnel (appelé stratégie ou politique, et qui est une fonction associant à l'état courant l'action à exécuter) optimal, en ce sens qu'il maximise la somme des récompenses au cours du temps.

II.3.6 Apprentissage automatique non supervisé

L'apprentissage non supervisé consiste à ne disposer que de données d'entrée (X) et pas de variables de sortie correspondantes.

L'objectif de l'apprentissage non supervisé est de modéliser la structure ou la distribution sous-jacente dans les données afin d'en apprendre davantage sur les données. Celles-ci sont appelées apprentissage non supervisé car, contrairement à l'apprentissage supervisé décrit avant, il n'y a pas de réponse correcte ni d'enseignant. Les algorithmes issus de l'apprentissage non supervisé sont laissés à leurs propres mécanismes pour découvrir et présenter la structure intéressante des données pour prendre une éventuelle décision.

Les problèmes d'apprentissage non supervisés peuvent se diviser en des problèmes de regroupement et d'association :

- **Mise en cluster** : Un problème de mise en cluster est l'endroit où nous souhaitons découvrir les regroupements inhérents dans les données, tels que le regroupement des clients en fonction du comportement d'achat.
- **Association :** Un problème d'apprentissage des règles d'association est l'endroit où nous souhaitons découvrir des règles décrivant une grande partie de nos données, telles que les personnes qui achètent X ont également tendance à acheter Y.

  Voici quelques exemples populaires d'algorithmes d'apprentissage non supervisé :

  - Isolation forest,
  - One class SVM.
  - LSTM.
  - K-Means

a) Méthode "forêt d'isolement (Isolation Forest)"

L'algorithme "forêt d'isolement" utilise le fait que les observations anormales sont peu nombreuses et significativement différentes des observations "normales". La forêt est construite sur la base d'arbres de décision où chacun des arbres ayant accès à un sous-échantillon des données de formation. Afin de créer une branche dans l'arborescence, une fonction aléatoire est d'abord sélectionnée. Ensuite, une valeur de partage aléatoire (entre les valeurs minimale et maximale) est choisie pour cette fonction. Si l'observation donnée a une valeur inférieure à cette caractéristique, celle qui est sélectionnée suit la branche gauche, sinon la branche droite [12]. Le partitionnement aléatoire produit des chemins sensiblement plus courts pour les anomalies. Par conséquent, lorsqu'une forêt composée d'arbres aléatoires produit collectivement des longueurs de chemin plus courtes pour des échantillons particuliers, il est fort probable que ce soient des anomalies. Ce processus se poursuit jusqu'à ce qu'un seul point soit isolé ou que la profondeur maximale spécifiée soit atteinte.

b) Méthode "Support Vector Machine"

Les machines à vecteurs de support (SVM[5]) sont une famille de classificateurs non supervisés qui ont été introduits dans les années 1990 par Vapnik et qui depuis sont devenus parmi les modèles les plus utilisés pour la classification et la régression [13].

[5] SVM : Support Vector Machines - en anglais

Les SVM cherchent à séparer deux groupes d'instances par un hyperplan de marge maximale. Un tel hyperplan est considéré comme un séparateur optimal qui aura une meilleure capacité à généraliser et à classifier les nouveaux exemples inconnus. La figure 3 montre comment sont séparés les groupes.

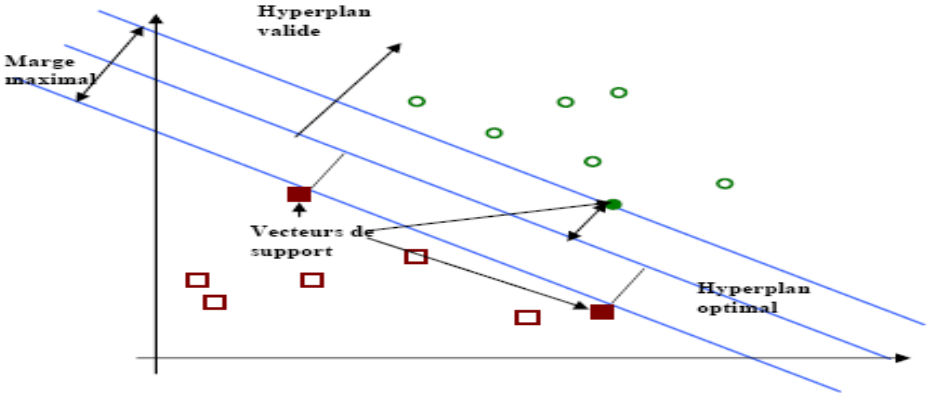


Figure 3: Exemple de marge maximal (hyperplan valide)

L'algorithme de la machine à vecteurs de support a pour objectif de trouver un hyperplan dans un espace à N dimensions où N représente le nombre d'attributs appelés aussi caractéristiques qui classe les points de données de manière différente [14].

c) Machines à vecteurs de support à une classe

Scholkopf et al ont proposé une extension des SVM permettant d'estimer le support de la distribution d'une loi de probabilité, appelée SVM à 1-classe. Pour cela, ils avaient proposé un algorithme qui peut estimer une fonction f dans la valeur est positive dans une région de l'espace d'observation et négative en dehors de cet espace. Par ailleurs dans cette nouvelle extension des SVM, plus le volume de cette région est petit, tout en garantissant une probabilité qu'une observation dans ce volume appartienne, meilleure est la description.

Les approches 1-classe ont de nombreuses applications comme la détection des points aberrants appelés aussi outliers, la détection d'anomalies ou encore la détection de nouveauté.

Nous trouvons dans cette variante de SVM une fonction de décision qui peut être non linéaire et basée sur les noyaux. La fonction de décision est faible dans le nombre d'échantillons, ce qui signifie essentiellement que nous avons très peu d'échantillons nous permettant de séparer ceux qui sont en dehors de la région normale de ceux qui sont à l'intérieur. L'approche SVM 1-classe se base sur l'idée des régions de densité où les points situés à l'intérieur d'une certaine région sont qualifiés de normaux, tandis que les points de données situés à l'extérieur de ces régions sont qualifiés d'anomalie. Elles sont meilleures que la densité du noyau, car celles-ci ne fonctionnent pas bien dans

les très grandes dimensions alors que One-Class SVM estime une fonction de décision qui fonctionne bien, même dans les grandes dimensions. Par conséquent, contrairement à la densité de noyau qui estime la densité, l'approche SVM 1-classe fournit une fonction de décision qui nous indique simplement que si la valeur du point de données est suffisamment basse, elle peut être considérée comme une anomalie ou non.

Dans ce type de SVM, le paramètre le plus fondamental et le plus important que nous utilisons est la valeur seuil, c'est-à-dire la fraction de données que nous supposons contaminées. Si nous fixons notre seuil à 50%, la moitié de nos données sera considérée comme une anomalie. Cependant, si nous prenons un nombre plus décent, disons 5%, alors 5% de nos données seront considérées comme une anomalie.

d) Long Short-Serm Memory (LSTM)

L'intérêt des réseaux de neurones avec mémoire (RNN) réside dans leur capacité à exploiter l'information contextuelle pour passer d'une séquence d'entrée à une séquence de sortie qui soit le plus proche possible de la séquence cible. Cependant, pour les RNN standards l'apprentissage peut se révéler difficile et le contexte réellement exploité très local. Le problème vient du fait qu'un vecteur d'entrée ne peut avoir une influence sur les décisions futures qu'au travers des liens récurrents via la multiplication répétée par la matrice V et l'application répétée de la fonction d'activation. Par conséquent, cette influence décroit ou augmente exponentiellement au fur et à mesure qu'on avance dans la séquence.

Un simple RNN avec mémoire appelé aussi réseau de neurones récurrent se construit juste en prenant la couche sortie de la précédente étape et la concaténer avec l'entrée de l'étape courante. Tout cela dans le but d'avoir la prédiction de l'étape courante, d'où le nom de récurrent pour la récurrence.

Le problème avec le RNN est le fait de garder en mémoire toutes les informations de chaque instant si l'on veut qu'il soit assez flexible. Par exemple dans le texte "J'ai grandi en Algérie ...(2000mots)... Je parle couramment xxxx". On voudrait prédire le xxxx. Le plus logique est xxxx soit égal à "Arabe ou Tamazighte". Mais pour qu'un RNN trouve que xxxx est égal à arabe ou tamazighte, il doit mémoriser 2000 instants, ce qui est conséquent. Tout cela va créer un très profond réseau de neurones au niveau de la récurrence (voir figure 4) et cela pénalise l'apprentissage en matière de mémoire (de l'ordinateur) et de temps d'apprentissage.

Comme solution, Hochreiter et Schmidhuber [15] ont proposé une solution appelée LSTM (Long-Short Term Memory Cell) pour pallier les limitations des RNN. Le LSTM est juste une autre forme de RNN. Le LSTM est conçu dans le but de supporter les problèmes aux longs termes de

dépendances parce que sa plus grande particularité est de mémoriser beaucoup d'informations. Dans une couche LSTM, on a quatre neurones que l'on appelle "Gate" alors que dans un RNN on a un seul neurone. Ces neurones de LSTM ont chacun un rôle et interagissent entre eux de manière spécifique.

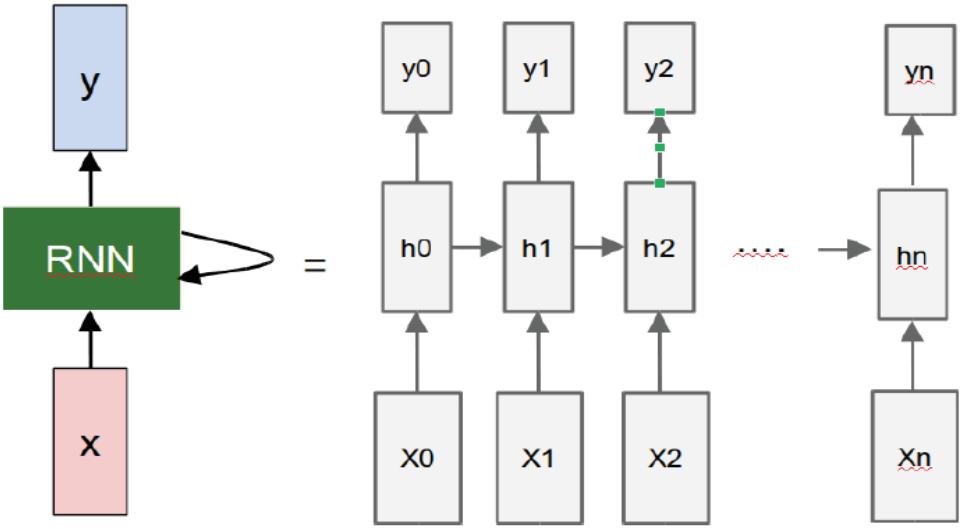


Figure 4: Exemple de réseau de neurones récurrents

e) L'approche K-Means

L**e** Clustering est l'une des techniques d'analyse de données exploratoires les plus courantes utilisées pour obtenir une intuition sur la structure des données. Cela peut être défini comme la tâche d'identifier les sous-groupes dans les données, de sorte que les points de données d'un même sous-groupe appelé aussi cluster soient très similaires, alors que les points de données de différents clusters sont très différents. Par ailleurs, la mise en clusters est considérée comme une méthode d'apprentissage non supervisée, car nous n'avons pas la vérité sur le terrain pour comparer le résultat de l'algorithme de mise en clusters aux libellés réels pour évaluer ses performances. Nous voulons seulement essayer d'étudier la structure des données en regroupant les points de données en sous-groupes distincts.

L'algorithme K-Means est un algorithme itératif qui tente de partitionner les données en *K* sous-groupes distincts prédéfinis, ne se chevauchant pas, dans lesquels chaque point de données appartient à un seul groupe. Il essaie de rendre les points de données inter-cluster aussi semblables que possible tout en gardant les clusters aussi différents (aussi loin que possible). Il attribue des points de données à un cluster de sorte que la somme de la distance au carré entre les points de données et le centre de gravité du cluster (moyenne arithmétique de tous les points de données appartenant à ce cluster) soit minimale. Moins il y a de variations dans les clusters, plus les points de données sont homogènes (similaires) dans le même cluster. Cependant, l'inconvénient de cette approche est que

nous devons dès le début fixer le nombre de clusters et les centres de ces clusters sont choisis aléatoirement.

II.4 Conclusion

Dans ce chapitre, nous avons présenté les techniques les plus répandues d'apprentissage automatique pour connaitre leurs avantages et leurs inconvénients en fonction de l'espace de données fournies. Cette présentation nous a permis de se focaliser un peu sur l'apprentissage non supervisé puisqu'on ne dispose pas de base de données réelles pour traiter la problématique abordée dans le cadre de ce projet de fin d'études.

Le chapitre suivant présente les outils utilisés pour traiter le problème de détection d'anomalies dans la consommation de l'électricité.

Chapitre III
Outils logiciels utilisés pour le développement de l'application

Chapitre III
Outils logiciels utilisés pour le développement de l'application

III.1 Introduction

L'apprentissage automatique, comme son nom l'indique, est la technique de la programmation d'un ordinateur grâce auquel il peut apprendre à partir de différents types de données. Une définition plus générale donnée par Arthur Samuel est la suivante : "L'apprentissage automatique est le domaine d'étude qui donne aux ordinateurs la capacité d'apprendre sans être explicitement programmé". Il est généralement utilisé pour résoudre divers types de problèmes dans divers domaines.

L'apprentissage automatique pourra être programmé sur ordinateurs en utilisant des plateformes bien spécifiques pour alléger la tâche aux chercheurs. Par exemple, Anaconda est une distribution qui inclue tous les paquets Python les plus courants, ainsi que de nombreux paquets liés à l'analyse de données et au Big Data. Python possède de nombreuses librairies, utilisées dans tous les domaines telles que Pandas**,** Matplotlib ou encore Numpy pour tester, explorer les données et les analyser.

Pour développer l'application dans le cadre de notre projet, nous avons utilisé la plateforme analytique KNIME et le langage Python. Nous décrivons dans ce chapitre les différents outils logiciels utilisés pour le développement de notre application.

III.2 Outils logiciels utilisés

III.2.1 Les bibliothèques Python pour l'apprentissage automatique

Les bibliothèques Python utilisées dans le Machine Learning dans notre cas sont les suivantes [16]:

- Numpy
- Scipy
- Scikit-learn

- Pandas
- Matplotlib
- Plotly

a) La bibliothèque Numpy

NumPy est une bibliothèque python très populaire pour le traitement de matrices et de tableaux multidimensionnels à l'aide d'un grand nombre de fonctions mathématiques de haut niveau. Cette bibliothèque 'est très utile pour les calculs scientifiques fondamentaux en Machine Learning. Elle est particulièrement utile pour l'algèbre linéaire, la transformée de Fourier et la génération des nombres aléatoires.

NumPy est le package fondamental du calcul scientifique avec Python. Il est principalement utilisé pour résoudre **des** problèmes matriciels dont l'intégration se fait syntaxiquement comme suit :

import numpy as np

L'emploi de raccourcis (ici np plutôt que numpy) permet de faciliter l'écriture des appels des fonctions de la librairie et le type de base dans NumPy est le tableau unidimensionnel ou multidimensionnel composé d'éléments de même type.

b) La bibliothèque SciPy

SciPy est une bibliothèque très populaire parmi les passionnés d'apprentissage automatique car elle contient différents modules d'optimisation, d'algèbre linéaire, d'intégration et de statistiques. SciPy est l'un des principaux packages qui composent la pile SciPy. SciPy est également très utile pour la manipulation d'images.

La fonctionnalité principale de la bibliothèque SciPy est basée sur NumPy, et ses tableaux. Ce package utilise beaucoup le package NumPy avec des modules pour la programmation scientifique, incluant l'algèbre linéaire, le calcul intégral (calcul différentiel), la résolution d'équation différentielle ordinaire et le traitement du signal.

c) La biblithèque Skikit-learn

Scikit-learn est une bibliothèque libre de Python destinée à l'apprentissage automatique. Elle est développée par de nombreux contributeurs notamment dans le monde académique par des instituts français d'enseignement supérieur et de recherche comme INRIA et Télécom ParisThec. Elle comprend notamment des fonctions pour estimer des forêts aléatoires, des régressions

logistiques, des algorithmes de classification, et machines à vecteurs de support. Elle est conçue pour s'harmoniser avec d'autres bibliothèques libres Python, notamment Numpy et SciPy.

from sklearn import datasets

Scikit-learn présente une interface concise et cohérente avec les algorithmes d'apprentissage automatique courants, facilitant ainsi l'introduction du ML dans les systèmes de production. La bibliothèque combine un code de qualité et une bonne documentation, une facilité d'utilisation et hautes performances. Elle constitue de facto le standard de l'industrie pour l'apprentissage automatique avec Python.

d) La bibliothèque Pandas

Pandas est une bibliothèque Python populaire pour l'analyse de données. Cette bibliothèque n'est pas directement liée à l'apprentissage automatique. Comme nous savons que le jeu de données doit être préparé avant la formation. Dans ce cas, les pandas sont pratiques car ils ont été développés spécifiquement pour l'extraction et la préparation de données. Le package Pandas fournit des structures de données de haut niveau et de nombreux outils pour l'analyse des données. Il fournit également de nombreuses méthodes intégrées pour tâtonner, combiner et filtrer les données.

import pandas as pd

Pandas est un package Python conçu pour fonctionner avec des données "étiquetées" et "relationnelles" simples et intuitives. Les modules de pandas constituent un outil parfait pour la gestion de données. Il a été conçu pour une manipulation, une agrégation et une visualisation rapides et faciles des données.

Dans la bibliothèque Pandas, il y a deux structures de données principales :

Tableau 2 : "Série" unidimensionnelle

Series	
A	X0
B	X1
C	X2
D	X3

Tableau 3:"Data Frames" en deux dimensions

DataFrame				
	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3

Par exemple, lorsque nous souhaitons recevoir un nouveau Dataframe à partir de ces deux types de structures, vous recevez ce type de fichier en ajoutant une seule ligne à un DataFrame en transmettant une série:

Tableau 4: Transformation de "DataFrame" en "Série"

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3
4	X0	X1	X2	X3

Voici une petite liste de choses que nous pouvons faire avec les Pandas :

- Supprimer et ajouter facilement des colonnes de DataFrame
- Convertir des structures de données en objets DataFrame
- Traiter les données manquantes, représentées en tant que NaN
- Regroupement puissant par fonctionnalité

e) La bibliothèque Matpoltlib

Matpoltlib est une bibliothèque Python très populaire pour la visualisation de données. Comme les Pandas, il n'est pas directement lié à l'apprentissage automatique. Cela s'avère particulièrement utile lorsqu'un programmeur souhaite visualiser les modèles dans les données. C'est une bibliothèque de tracé 2D et 3D utilisée pour créer des graphiques et des tracés 2D et 3D. Un module appelé pyplot facilite le traçage des programmeurs car il offre des fonctionnalités permettant de contrôler les styles de trait, les propriétés de police, les axes de formatage, etc.

import matplotlib.pyplot as plt

Avec ce package, nous pouvons réaliser à peu près toutes les visualisations :

- Tracés linéaires ;
- Nuages de points;
- Diagrammes à barres et histogrammes;
- Camemberts;
- Parcelles de tiges;
- Tracés de contour;
- Parcelles de carquois;
- Spectrogrammes.

f) La bibliothèque Plotly

Il s'agit plutôt d'une boîte à outils Web pour la création de visualisations, exposant les API à certains langages de programmation (parmi lesquels Python). Il existe un certain nombre de graphiques prêts à l'emploi sur le site intrigue.ly. Pour utiliser Plotly, nous devons configurer notre clé API. Les graphiques seront traités côté serveur et seront postés sur Internet, mais il existe un moyen de l'éviter.

import plotly.plotly as py

II.2.2 La plateforme KNIME

"KNIME Analytics Platform" est basé sur un logiciel open source permettant de créer des applications et des services de science des données. Intuitif, ouvert et intégrant en permanence les nouveaux développements, KNIME facilite la compréhension des données et la conception de workflows de science des données et de composants réutilisables.

KNIME est un atelier graphique convivial pour l'ensemble du processus d'analyse : accès aux données, transformation des données, investigation initiale, puissante analyse prédictive, visualisation et rapport. La plateforme d'intégration ouverte fournit plus de 1000 modules (nœuds), y compris ceux de la communauté KNIME et de son vaste réseau de partenaires. La plateforme KNIME a été conçue pour des applications générales. La configuration de base contient donc des fonctionnalités communes à de nombreux domaines. De plus, il existe de nombreuses extensions en informatique et en bio-informatique qui sont utiles en particulier dans la découverte de médicaments en phase précoce, mais peuvent être appliquées partout où une gestion ou une analyse de la structure chimique est nécessaire.

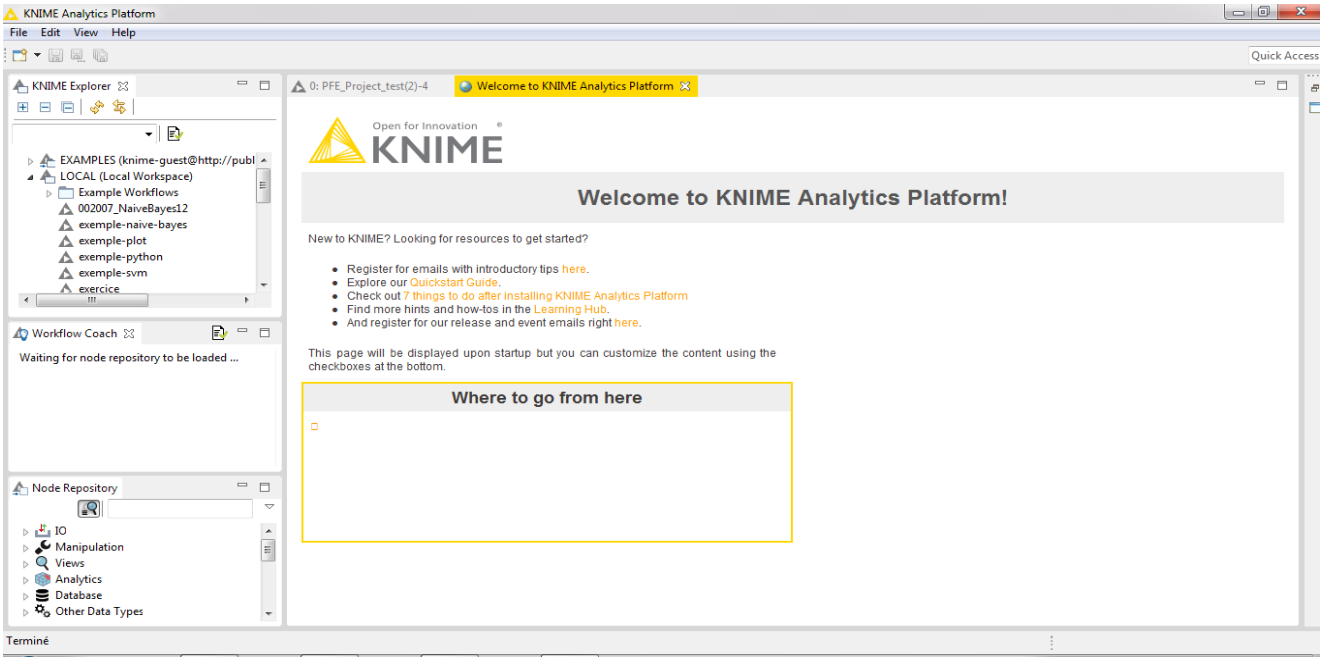


Figure 5 : Interface de la plateforme KNIME

KNIME permet de :

- Créer des flux de travail visuels avec une interface graphique intuitive de style glisser-déposer, sans nécessiter de codage.
- Fusionner des outils de différents domaines avec des nœuds KNIME natifs dans un seul flux de travail, notamment des scripts dans R & Python, l'apprentissage automatique ou des connecteurs vers Apache Spark.
- Choisir parmi plus de 2000 modules ("nœuds") pour construire votre flux de travail. Modéliser chaque étape de l'analyse, contrôler le flux de données.
- Sélectionner l'un des centaines d'exemples de flux de travaux disponibles au public ou utiliser le coach de flux de travaux intégré pour nous guider dans la création de notre flux de travaux.

Les flux de travail KNIME peuvent être utilisés comme des ensembles de données pour créer des modèles de rapport pouvant être exportés vers des formats de document tels que doc, ppt, xls, pdf et autres. Les autres fonctionnalités de KNIME sont :

- L'architecture de base de KNIME permet le traitement de gros volumes de données qui ne sont limités que par l'espace disponible sur le disque dur (et non par la RAM disponible). KNIME, par exemple, permet d'analyser 300 millions d'adresses de clients, 20 millions d'images de cellules et 10 millions de structures moléculaires.

- Des plugins supplémentaires permettent l'intégration de méthodes d'exploration de texte, d'exploration d'images, ainsi que l'analyse de séries chronologiques.

II.2.3 La base de données (dataset)

Les données manipulées sont extraites du Smart *, une infrastructure de collecte de données qui enregistre des données à partir d'une variété de capteurs déployés dans de vraies maisons. L'infrastructure prend en charge la collecte de données en interrogeant des capteurs individuels et transfert des données des capteurs à une passerelle serveur, qui exécute les outils logiciels.

L'infrastructure de collecte de données fournit une interface Web pour configurer les appareils dans chaque maison et contrôler le processus de collecte de données.

a) SMART * OPEN SET DE DONNÉES

La version initiale comprend deux ensembles de données :

(i) une solution haute résolution ensemble de données provenant de trois foyers,

(ii) une résolution de données inférieure ensemble de 400 maisons. Nous nous référons à l'ancien comme l'UMass Smart *
Home Data Set et ce dernier en tant que données UMass Smart * Microgrid

b) Jeu de données Smart * Home

Home A : est une maison de deux étages de 1 700 pieds carrés avec trois résidences à temps plein. La maison dispose d'un total de huit chambres, y compris son sous-sol. Le niveau principal comprend un salon, une chambre, une cuisine et salle de bains, tandis que la deuxième comprend deux chambres et une salle de bains.

La maison A est une maison la plus profondément instrumentée dans laquelle il y a l'utilisation de capteurs installés dans le panneau d'alimentation, la collection des données d'électricité toutes les secondes pour toute la maison, ainsi que chaque circuit. Il y a environ 30 des 35 interrupteurs muraux à remplacer dans la maison avec unités qui transmettent on-offdim événements pour les commutateurs sur la ligne électrique vers un serveur de passerelle.

III.3 Description des méthodes de ML utilisées

III.3.1 Méthode 1 : Isolation Forest

Forêt d'isolation est une méthode qui permet la détection d'anomalies. Elle possède des caractéristiques remarquables qui ont motivé son choix pour notre étude. Il est simple, repose sur

très peu d'hypothèses, peut très facilement être adaptée aux paradigmes de programmation distribuée, et a prouvé son efficacité dans la détection d'anomalies.

L'algorithme de la forêt d'isolation débute par une phase d'apprentissage non supervisé, qui aboutit à la construction d'arbres binaires. Considérant un lot de données à traiter constitué de vecteurs de caractéristiques possédant N dimensions, un sous échantillon aléatoire de taille constante est sélectionné pour la construction de chaque arbre binaire.

La génération des branches s'effectue en sélectionnant au hasard selon une loi uniforme : une des dimensions $x_i \in \{x_1, x_2, \ldots., x_n\}$ de la matrice de données, puis une valeur seuil v, choisie également au hasard entre le minimum et le maximum observés pour cette caractéristique. Si un vecteur de caractéristiques possède une valeur inférieure ou égale à v, il est passé à la branche droite, sinon il est passé à la branche gauche.

L'Alimentation du jeu de données et le calcul du score d'anomalie se fait comme suit :

- Introduire chaque point de données dans un modèle de forêt formé pour chaque arbre
- Le score d'anomalie est défini comme :

$$Anomaly\ Score(S) = 2^{\frac{-E(h(k,m,N))}{C(n)}}$$

Où $c(n) = 2(ln(n-1) + 0.5772156649) - 2\left(\frac{n-1}{n}\right)$

Où n est le nombre de points de données dans une sélection

Où $E(h(k,m,N)) = \frac{\sum_{i=1}^{N} \begin{cases} if\ \ k==1, \sum_{j=1}^{M} 1 \\ else,\ \ \sum_{j=1}^{M} 1 + c(k) \end{cases}}{N}$

Où N : le nombre total d'arbres, M ; le nombre total de scissions binaire et k : le nombre total de points de données dans le nœud final.

III.3.2 Méthode 2 : One-classs SVM

Ce sont des algorithmes d'apprentissage initialement construits pour la classification binaire. L'idée est de rechercher une règle de décision basée sur une séparation par hyperplan de marge optimale.

Le principe de l'algorithme basé sur la méthode 1-classe SVM est d'intégrer lors de la phase d'apprentissage une estimation de sa complexité pour limiter le phénomène d'over-fitting. L'algorithme s'appuie principalement sur trois astuces pour obtenir de très bonnes performances tant en qualité de prédiction qu'en complexité de calcul.

On cherche l'hyperplan comme solution d'un problème d'optimisation sous-contrainte. La fonction à optimiser intègre un terme de qualité de prédiction et un terme de complexité du modèle

Le passage à la recherche de surfaces séparatrices non linéaires est introduit en utilisant un noyau "kernel" qui code une transformation non linéaire des données. Numériquement, toutes les équations s'obtiennent en fonction de certains produits scalaires utilisant le noyau et certains points de la base de données.

III.3.3 Méthode 3 : K-Means

La méthode K-Means est basée sur un algorithme itératif qui minimise la somme des distances entre chaque individu et le centroïd. Le choix initial des centroïds conditionne le résultat final**.**

Admettant un nuage d'un ensemble de points, K-Means change les points de chaque cluster jusqu'à ce que la somme ne puisse plus diminuer. Le résultat est un ensemble de clusters compacts et clairement séparés, sous réserve de choisir la bonne valeur du nombre de clusters et l'emplacement des centroids.

Schéma algorithmique de K-Means

Entrée :

- K le nombre de cluster à former
- Le Training Set (matrice de données)

DEBUT

- Choisir aléatoirement K points (une ligne de la matrice de données). Ces points sont les centres des clusters (nommés centroïds).

REPETER

- Affecter chaque point (élément de la matrice de donnée) au groupe dont il est le plus proche au son centre,
- Recalculer le centre de chaque cluster et modifier le centroid

JUSQU'A CONVERGENCE

OU (stabilisation de l'**inertie totale** de la population)

FIN ALGORITHME

Lors de la définition de l'algorithme, quand on parle de "point", c'est un point au sens "donnée/data" qui se trouve dans un espace vectoriel de dimension n. Avec n : le nombre de colonnes de la matrice de données.

La convergence de l'algorithme K-Means peut être l'une des conditions suivantes :

- Un nombre d'itérations fixé à l'avance, dans ce cas, K-Means effectuera les itérations et s'arrêtera peu importe la forme de clusters composés.
- Stabilisation des centres de clusters (les centroids ne bougent plus lors des itérations).

L'affectation d'un point à un cluster se fait en fonction de la distance de ce point par rapport aux différents centroids. Par ailleurs, ce point se fera affecté à un cluster s'il est plus proche de son centroid (distance minimale). Finalement, la distance entre deux points dans le cas de K-Means se calcule par différentes méthodes telles que la distance Euclidienne et la distance de Manhattan.

III.4 Conclusion

Dans ce chapitre, nous avons présenté les différents outils logiciels nécessaires pour le développement de notre application qui consiste à détecter les anomalies dans la consommation de l'électricité au sein des bâtiments tels les bibliothèques de Python et la plateforme KNIME.

Le chapitre suivant fait l'objet de l'application développée en montrant comment ces outils ont été impliqués.

Chapitre IV
Détection d'anomalies dans la consommation d'électricité par ML

Chapitre IV
Détection d'anomalies dans la consommation d'électricité par ML

IV.1 Introduction

Dans ce chapitre, nous présentons notre application qui consiste à détecter les anomalies (outliers) dans la consommation de l'électricité en utilisant trois méthodes différentes : Isolation Forest, 1-class SVM et K-Means. Ces méthodes de détection se basent sur un apprentissage non supervisé.

Nous avons pris comme environnement de test un habitat dans lequel il y a un ensemble d'équipements tels que une machine à laver, une lave vaisselle, un chauffage et un climatiseur, etc. Par ailleurs, il existe des compteurs intelligents qui permettent de connaitre l'équipement qui est actif à un instant donné en fonction de la consommation de l'électricité. Néanmoins, quand il existe des équipements qui consomment la même quantité d'électicité dans ce cas nous ne pouvons pas distinguer entre les deux et connaitre celui qui est actif. A cet effet, nous avons proposé de faire doter ces équipements de capteurs et par suite quand un équipement est actif ceci peut être signalé par son capteur sous-jacent.

Dans ce que suit, nous présentons l'environnement de travail, l'évaluation de la consommation d'électricité dans un habitat à travaers trois méthodes de classification et nous comparons les performances de chacune en termes de fiabilité dans la détection des anomalies dans la consommation d'électricité. Cette fiabilité est mis en œuvre en faisant varier un paramètre de précision représenté par *outlier_fraction.*

IV.2 Environnement du développement

Nous présentons dans ce chapitre, la détection des anomalies de données de consommation énergétique dans les habitats. Ces habitats contiennent un ensemble d'équipements tels que machine à laver, lave-vaisselle, climatiseur, etc … Pour cela, nous avons utilisé un ensemble de données pour décrire la consommation d'énergie et trois méthodes pour détecter les éventuelles anomalies dans la consommation de l'électricité. La figure 6 représente le fonctionnement de l'application que nous avons développée.

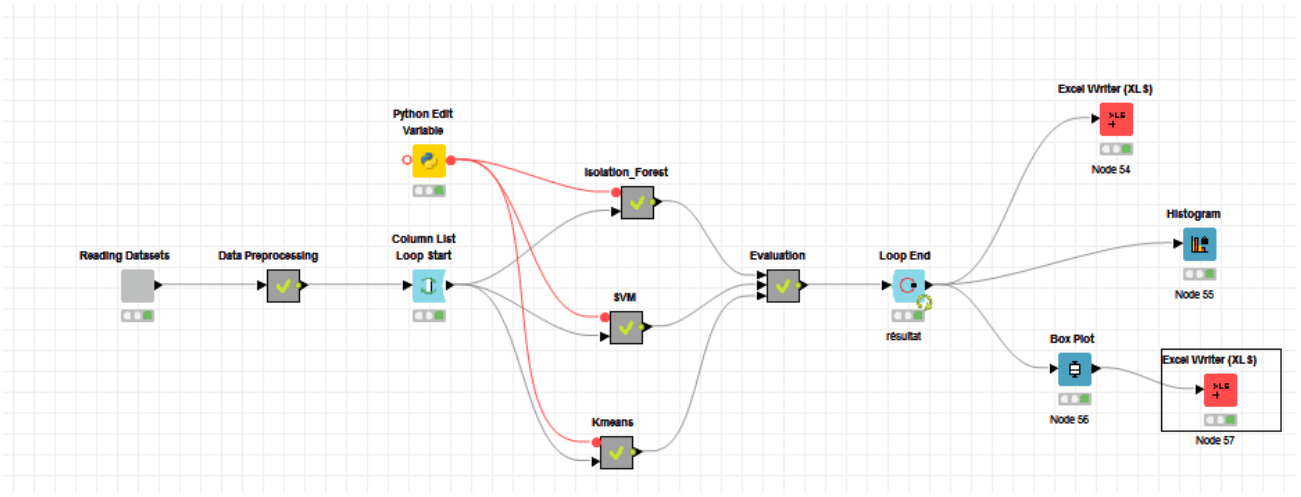


Figure 6: Représentation du modèle

IV.2.1 Accès aux données

Pour prévoir la consommation d'électricité, il faut connaître les variables d'entrée liées à la consommation. Les variables doivent être extraites de plusieurs sources de données. Les données du bâtiment sont souvent gardées privées par les propriétaires.

Les traces de consommation sont disponibles pour plusieurs années au format CSV et incluent un résumé pour chaque fichier qui est accessible au public sur le Web : http://lass.cs.umass.edu/projects/smart.

Les données de chaque bâtiment se composent de la consommation d'électricité à intervalle de temps constant. La fréquence commune la plus basse dans les données est 1 heure, ce qui signifie qu'il y a 24 mesures chaque jour.

Dans le contexte de la détection des anomalies, cela devrait être suffisant, car des anomalies plus courtes moins d'une heure ont peu d'influence sur la consommation totale d'électricité.

Les données d'entrée

Nous commençons donc par décrire nos données.

Le jeu de données est extrait du site Umass [17,18].

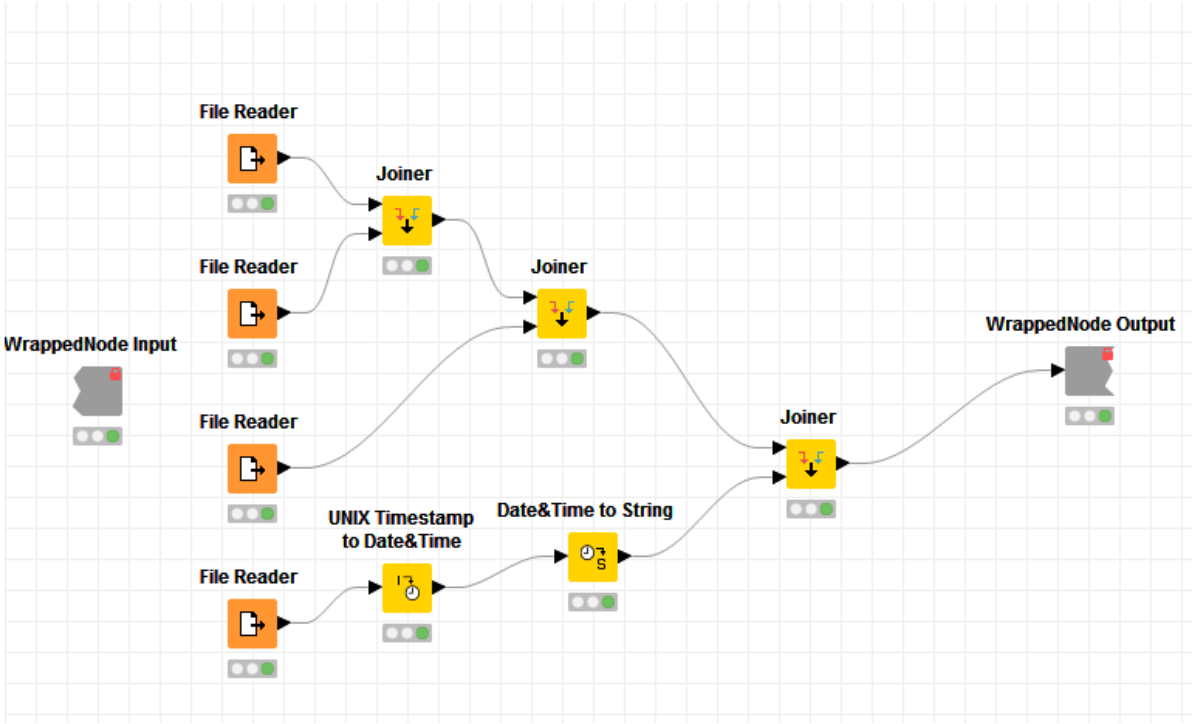


Figure 7: Données d'entrée

IV.2.2 Description des nœuds

Les données sont décrites sous forme d'un fichier CSV contenant les valeurs des différents équipements. Nous avons pris comme exemple le fichier F:\files\HomeA\2014.

Tableau 5 : Table d'entrée de données

Row ID	Date & Time	use [kW]	gen [kW]	Furnac...	CellarO...	Washin...	FridgeR...	Disposa...	Kitchen...	Bedroo...	Bedroo...	Master...	Master...	DuctHe...
Row0	2014-01-01 00:00:00	0	0	0.195	0.083	0.006	0.007	0.006	0.012	0.02	0.005	0.047	0.01	0.383
Row1	2014-01-01 00:30:00	0	0	0.182	0.036	0.006	0.094	0.005	0.005	0.021	0.009	0.052	0.01	0.276
Row2	2014-01-01 01:00:00	0	0	0.135	0.047	0.006	0.015	0.006	0.003	0.021	0.005	0.054	0.01	0.346
Row3	2014-01-01 01:30:00	0	0	0.182	0.071	0.006	0.082	0.005	0.003	0.021	0.005	0.051	0.01	0.31
Row4	2014-01-01 02:00:00	0	0	0.093	0.014	0.006	0.032	0.005	0.003	0.02	0.005	0.049	0.01	0.31
Row5	2014-01-01 02:30:00	0	0	0.131	0.071	0.006	0.055	0.005	0.003	0.02	0.005	0.049	0.01	0.341
Row6	2014-01-01 03:00:00	0	0	0.055	0.039	0.006	0.04	0.005	0.003	0.021	0.005	0.049	0.01	0.269
Row7	2014-01-01 03:30:00	0	0	0.082	0.044	0.006	0.049	0.006	0.003	0.021	0.005	0.049	0.01	0.378
Row8	2014-01-01 04:00:00	0	0	0.115	0.07	0.006	0.038	0.005	0.003	0.021	0.005	0.049	0.01	0.231
Row9	2014-01-01 04:30:00	0	0	0.034	0.007	0.006	0.052	0.006	0.006	0.021	0.005	0.049	0.01	0.418
Row10	2014-01-01 05:00:00	0	0	0.086	0.067	0.006	0.028	0.005	0.003	0.021	0.005	0.047	0.01	0.195
Row11	2014-01-01 05:30:00	0	0	0.089	0.051	0.006	0.06	0.006	0.003	0.02	0.005	0.047	0.011	0.463
Row12	2014-01-01 06:00:00	0	0	0.237	0.02	0.006	0.013	0.005	0.003	0.021	0.005	0.045	0.009	0.152
Row13	2014-01-01 06:30:00	0	0	0.69	0.078	0.006	0.081	0.006	0.003	0.021	0.005	0.046	0.011	0.481
Row14	2014-01-01 07:00:00	0	0	0.646	0.029	0.006	0.019	0.005	0.003	0.021	0.005	0.046	0.009	0.109
Row15	2014-01-01 07:30:00	0	0	0.573	0.045	0.006	0.075	0.006	0.003	0.021	0.007	0.046	0.011	0.513
Row16	2014-01-01 08:00:00	0	0	0.593	0.08	0.006	0.039	0.005	0.052	0.027	0.006	0.052	0.062	0.062
Row17	2014-01-01 08:30:00	0	0	0.521	0.01	0.006	0.056	0.006	0.254	0.027	0.005	0.051	0.011	0.555
Row18	2014-01-01 09:00:00	0	0	0.563	0.078	0.006	0.074	0.005	0.239	0.027	0.005	0.051	0.009	0.037
Row19	2014-01-01 09:30:00	0	0	0.601	0.042	0.006	0.042	0.006	0.003	0.022	0.019	0.051	0.047	0.575
Row20	2014-01-01 10:00:00	0	0	0.499	0.044	0.006	0.062	0.005	0.003	0.013	0.066	0.052	0.168	0.038
Row21	2014-01-01 10:30:00	0	0	0.138	0.073	0.094	0.052	0.054	0.003	0.012	0.005	0.047	0.011	0.567
Row22	2014-01-01 11:00:00	0	0	0.052	0.011	0.145	0.05	0.746	0.003	0.013	0.004	0.043	0.009	0.039
Row23	2014-01-01 11:30:00	0	0	0.063	0.082	0.114	0.085	0.353	0.003	0.014	0.005	0.04	0.011	0.571
Row24	2014-01-01 12:00:00	0	0	0.103	0.032	0.006	0.06	0.005	0.003	0.023	0.005	0.04	0.088	0.034
Row25	2014-01-01 12:30:00	0	0	0.055	0.054	0.005	0.022	0.006	0.003	0.023	0.005	0.04	0.022	0.558

D'après le tableau 5, on a trois fichiers décrivant les variations de chaque équipement par jour/heure.

Le quatrième nœud nous donne les différents facteurs influant sur les équipements comme s'est illustré par le tableau 6.

Tableau 6: Facteurs influents les équipements

Row ID	temper...	icon	humidity	visibility	summary	appare...	pressure	windSp...	cloudC...	time	windBe...	precipI...	dewPoint	precipP...
Row0	16.67	clear-night	0.53	10	Clear	3.95	1,022.69	11.23	0	1388552400	271	0	2.41	0
Row1	16.19	clear-night	0.52	10	Clear	4.18	1,022.58	9.92	0.02	1388556000	268	0	1.68	0
Row2	15.69	clear-night	0.55	10	Clear	4.41	1,023.61	8.72	0	1388559600	266	0	2.59	0
Row3	15.29	clear-night	0.58	10	Clear	4.29	1,024.31	8.25	0	1388563200	269	0	3.29	0
Row4	15.37	clear-night	0.6	10	Clear	5.12	1,024.51	7.38	0.06	1388566800	266	0	3.95	0
Row5	14.81	clear-night	0.65	9.88	Clear	4.37	1,024.83	7.47	0	1388570400	261	0	5.11	0
Row6	14.4	clear-night	0.66	9.9	Clear	3.9	1,025.59	7.43	0	1388574000	253	0	5.28	0
Row7	13.79	clear-night	0.67	9.95	Clear	4.45	1,026.09	6.1	0.17	1388577600	247	0	4.98	0
Row8	14.5	clear-day	0.69	9.67	Clear	6.35	1,026.79	5.14	0	1388581200	253	0	6.18	0
Row9	17.05	clear-day	0.7	8.97	Clear	11.24	1,027.79	3.7	0	1388584800	295	0	8.93	0
Row10	18.29	clear-day	0.65	9.29	Clear	7.31	1,028.24	9.14	0.21	1388588400	252	0	8.4	0
Row11	19.68	clear-day	0.57	10	Clear	10.2	1,028.07	7.54	0	1388592000	276	0	6.92	0
Row12	20.82	clear-day	0.54	10	Clear	10.85	1,027.53	8.53	0	1388595600	270	0	6.69	0
Row13	21.61	partly-cloud...	0.52	10	Partly Cloudy	12	1,027.25	8.26	0.53	1388599200	278	0	6.52	0
Row14	21.63	clear-day	0.51	9.56	Clear	11.84	1,027.15	8.53	?	1388602800	277	0	6.48	0
Row15	21.74	clear-day	0.52	10	Clear	13.65	1,027.4	6.39	?	1388606400	289	0	6.97	0
Row16	21.52	partly-cloud...	0.53	10	Mostly Cloudy	14.46	1,027.78	5.28	0.75	1388610000	294	0	6.89	0
Row17	19.73	clear-night	0.57	10	Clear	12.71	1,027.82	4.95	0	1388613600	285	0	7.11	0

Les nœuds de jointure permettent de fusionner les données en tenant compte de la même date-time comme référence.

IV.2.3 Filtrage des données

L'une des premières tâches de l'analyse de données consiste à extraire uniquement certains des enregistrements disponibles dans le jeu de données d'origine. Par exemple, il est courant d'extraire des données uniquement pour une période donnée, dans notre cas on a limité la période de 6 mois du 01/01/2014 au 30/06/2014, on a éliminé les données aberrantes dans les données d'enquête ; et on s'est débarrassé des enregistrements manquants dans les séries chronologiques dérivées de capteurs ; etc. La figure résume la préparation des données pour effectuer l'analyse du nouveau fichier résultant des différentes opérations citées précédemment.

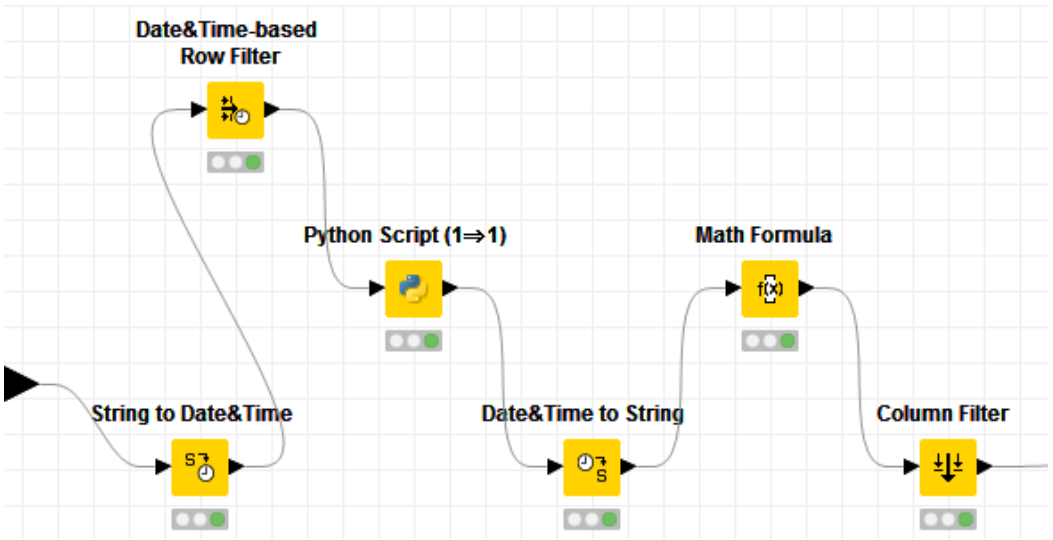


Figure 8: Préparation des données

IV.2.4 Analyse des données

C'est la partie la plus importante de tout projet d'analyse de données. Une fois les données filtrées et préparées correctement pour alimenter l'algorithme d'entraînement, il suffit de choisir l'algorithme d'apprentissage automatique à utiliser.

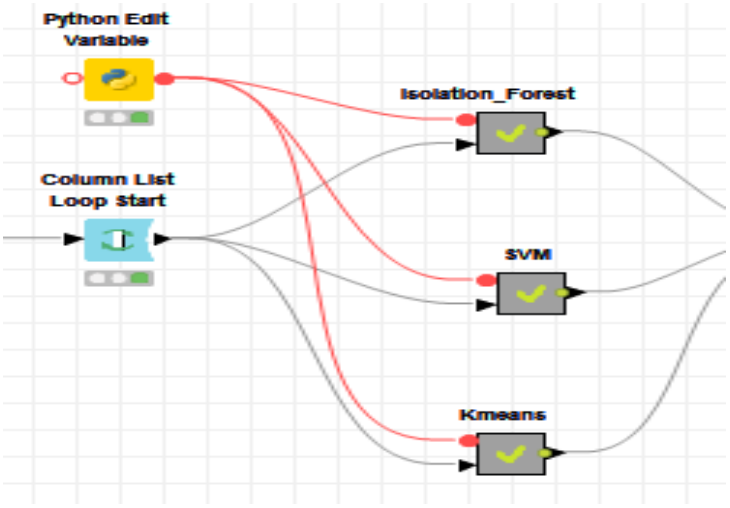


Figure 9: Les méthodes de ML utilisées

Tout d'abord, nous devons préciser les proportions de valeurs aberrantes (outlier_fraction).

IV.2.5 Edition des variables

Lors de l'application des trois modèles : IsolationForest (IF), One-Class-SVM (OCsvm) et KMeans, nous définissons les variables *outliers_fraction* qui indiquent la proportion de valeurs aberrantes dans l'ensemble de données est comme suit :

```
flow_variables['outlier_fraction_IF'] = 0.1
flow_variables['outlier_fraction_OCsvm'] = 0.1
flow_variables['outlier_fraction_KMeans'] = 0.2
```

Avant d'envoyer les données aux différentes méthodes pour les analyser, on introduit le nœud "*Column List Loop Start*". Ce nœud de démarrage de boucle effectue une itération sur une liste de colonnes de la table en entrée. Ces colonnes sont divisées en deux ensembles, les colonnes "exclues" seront toujours incluses dans toute itération, les colonnes "incluses" seront visibles une fois par itération.

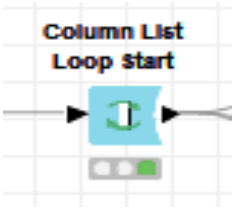


Figure 10: Nœud de démarrage de boucle

IV.3 Méthodes de ML utilisées

L'analyse prédictive consiste à analyser les données actuelles afin de faire des hypothèses sur des comportements futurs du système étudié. Nous nous servons des données que nous les possédons déjà pour extrapoler et deviner le comportement de nouveaux composants mais également l'évolution des composants déjà présents. Nous décidons ensuite de nous servir des résultats obtenus grâce à l'analyse descriptive, à savoir les huit groupes définis par les trois méthodes pour appliquer un modèle d'analyse prédictive à nos données.

Par ailleurs, la détection d'éventuelles anomalies permet de trouver les points dans les données qui ne correspondent pas au reste des données.

La liste des algorithmes non supervisés utilisés, issus des statistiques ou de la communauté de machine learning, prédisant des valeurs numériques, nécessitant des tendances temporelles et ces algorithmes exécutent les mêmes étapes suivantes :

- *Fit et predict (data)* effectue la détection des valeurs aberrantes sur les données et renvoie 1 pour la normale, -1 pour l'anomalie.
- On a normalisé les outliers pour toutes les méthodes par :

```
df['anomaly'].loc[df['anomaly'] == 1] = 0
df['anomaly'].loc[df['anomaly'] == -1] = 2
```

Enfin, nous visualisons les anomalies avec la vue série chronologique.

Dans cette étude, nous avons essayé de modifier le paramètre *Outlier-fraction* pour chaque algorithme ; nous avons utilisé deux valeurs 0.1 et 0.01. Nous décrivons dans ce que suit les algorithmes sous-jacents aux trois méthodes :

IV.3.1 Méthode 1 : Isolation Forest

L'algorithme de l'Isolation Forest isole les observations en sélectionnant de manière aléatoire une entité, puis en sélectionnant de manière aléatoire une valeur de partage entre les valeurs maximale et minimale de l'entité sélectionnée.

Le partitionnement récursif pouvant être représenté par une arborescence, le nombre de fractionnements requis pour isoler un échantillon est équivalent à la longueur du chemin d'accès du nœud racine au nœud de terminaison.

Cette longueur de chemin, calculée en moyenne sur une forêt de tels arbres aléatoires, est une mesure de la normalité et de notre fonction de décision.

Le partitionnement aléatoire produit des chemins sensiblement plus courts pour les anomalies. Par conséquent, lorsqu'une forêt composée d'arbres aléatoires produit collectivement des longueurs de chemin plus courtes pour des échantillons particuliers, il est fort probable que ce soient des anomalies.

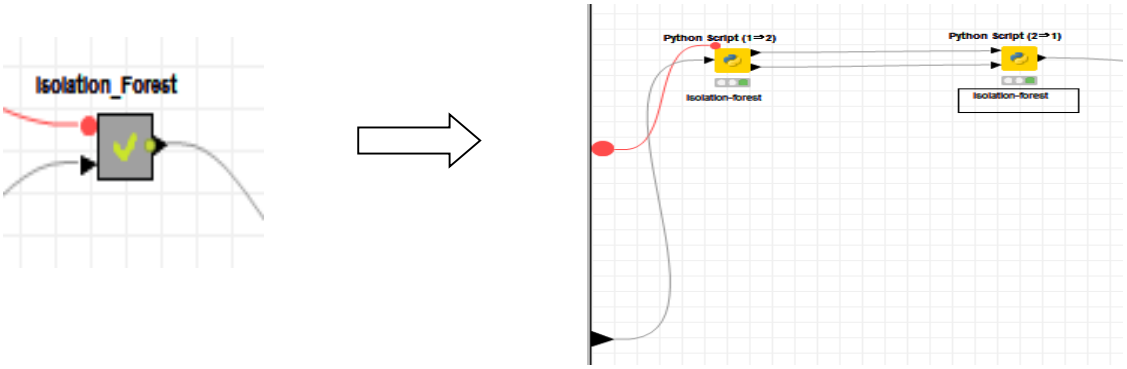
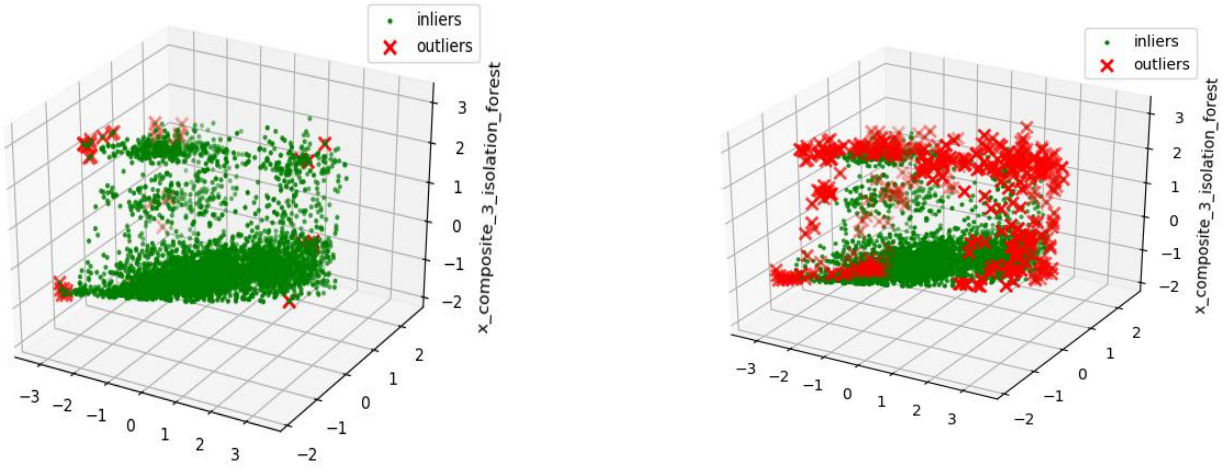


Figure 11: Méthode Isolation Forest

Nous avons maintenant 08 métriques sur lesquelles nous avons classé les anomalies en utilisant la méthode "la forêt d'isolation". Nous allons essayer de visualiser les résultats et de vérifier si la classification a un sens. Puis, nous normalisons et nous ajustons les métriques à une PCA pour réduire le nombre de dimensions, puis nous traçons les en 3D en soulignant les anomalies.

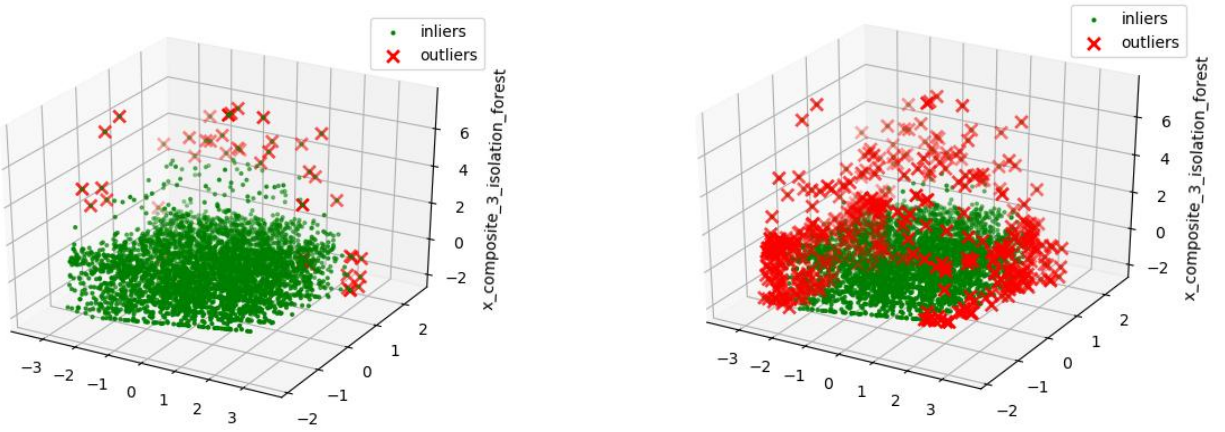
Le programme suivant (écrit en Python) représente les étapes pour analyser des données en utilisant la méthode "Isolation Forest". Les analyses sont faites avec deux valeurs du paramètre "*outliers_fraction*" : 0.1 et 0.01.

```
# Only use numeric columns
metrics_df=input_table.set_index('Date & Time')
metrics_df.reset_index(inplace=True)
metrics_df.fillna(0, inplace=True)
to_model_columns = metrics_df.columns[1:13]
from sklearn.ensemble import IsolationForest
clf = IsolationForest( contamination=flow_variables['outlier_fraction_IF'])
clf.fit(metrics_df[to_model_columns])
pred = clf.predict(metrics_df[to_model_columns])
metrics_df['anomaly'] = pred
outliers = metrics_df.loc[metrics_df['anomaly'] == -1]
outlier_index = list(outliers.index)
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from mpl_toolkits.mplot3d import Axes3D
pca = PCA(n_components=3)  # Reduce to k=3 dimensions
scaler = StandardScaler()
# normalize the metrics
X = scaler.fit_transform(metrics_df[to_model_columns])
X_reduce = pca.fit_transform(X)
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.set_zlabel("x_composite_3_isolation_forest  ")
# Plot the compressed data points
ax.scatter(X_reduce[:, 0], X_reduce[:, 1], zs=X_reduce[:, 2], s=4, lw=1, label="inliers", c="green")
# Plot x's for the ground truth outliers
ax.scatter(X_reduce[outlier_index, 0], X_reduce[outlier_index, 1], X_reduce[outlier_index, 2],
        lw=2, s=60, marker="x", c="red", label="outliers")
ax.legend()
plt.show()
```



(a) outliers_fraction=0.01 (b) outliers_fraction=0.1

Figure 12: Représentation 3D de Furnace HRV [kw] – Isolation Forest



(a) outliers_fraction=0.01 (b) outliers_fraction=0.1

Figure 13: Représentation 3D de Fridge Range [kw] – Isolation Forest

Maintenant que nous voyons dans la représentation 3D, les points d'anomalie sont souvent éloignés du groupe des points normaux. Nous avons aussi déterminé le comportement anormal au niveau du cas d'utilisation. Par ailleurs, les anomalies identifiées par l'algorithme doivent avoir un sens lorsqu'elles sont visualisées par un utilisateur pour agir en conséquence.

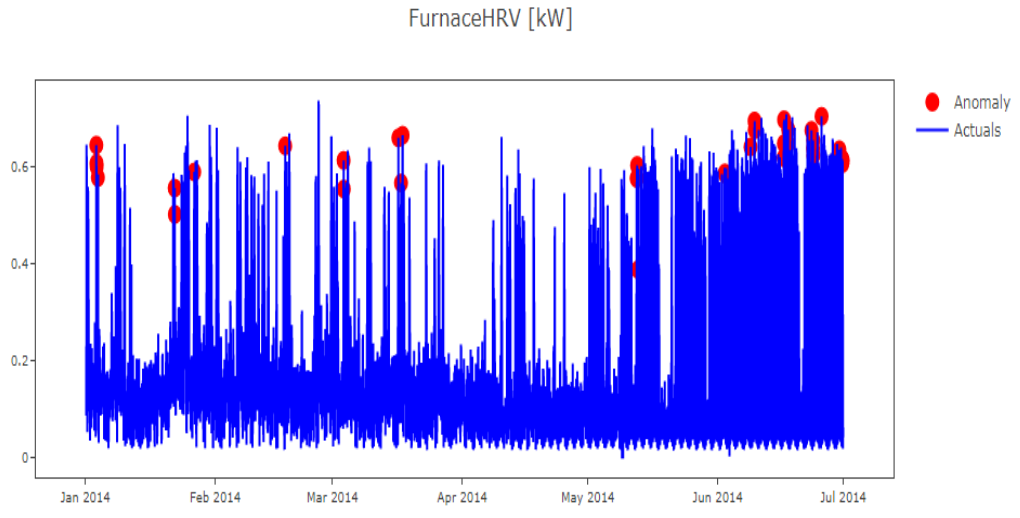
Dans ce programme, il existe une fonction qui permet de créer un graphique réel sur une série chronologique avec des points d'anomalie mis en évidence. Également un tableau qui fournit les données réelles, la modification et la mise en forme conditionnelle en fonction des anomalies. Il existe aussi une fonction d'assistance permettant de rechercher le pourcentage de variation et de classer les anomalies en fonction de leur gravité. En plus, la fonction de prédiction classe les données en tant

qu'anomalies sur la base des résultats de la fonction de décision lors du franchissement d'un seuil. Les 12 premiers quantiles sont des anomalies identifiées (gravité élevée). En outre, sur la base de la fonction de décision, nous identifions ici les 12–24 points de quantiles et les classons comme anomalies de faible gravité.

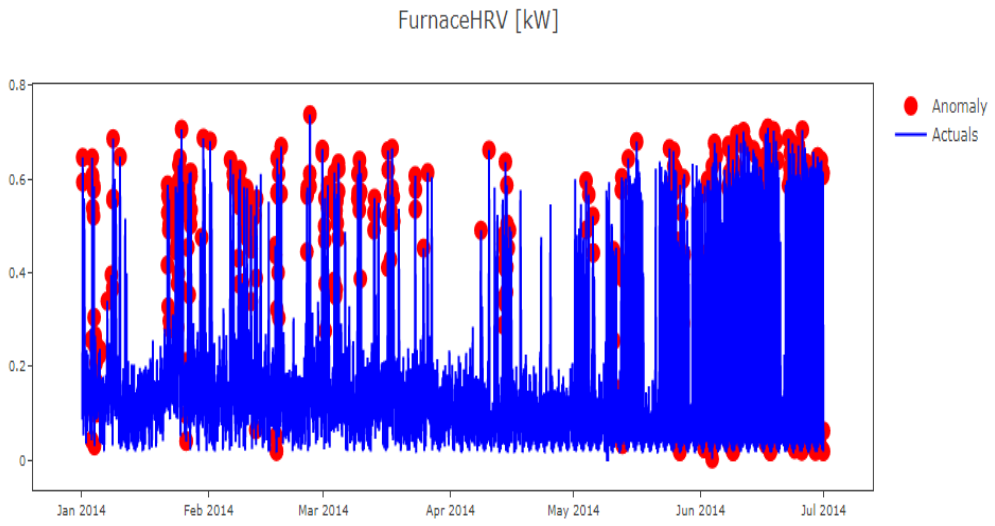
```
def classify_anomalies(df, metric_name):
    df['metric_name'] = metric_name
    df = df.sort_values(by='Date & Time', ascending=False)
    # Shift actuals by one timestamp to find the percentage chage between current and previous data point
    df['shift'] = df['actuals'].shift(-1)
    df['percentage_change'] = ((df['actuals'] - df['shift']) / df['actuals']) * 100
    # Categorise anomalies as 0-no anomaly, 1- low anomaly , 2 - high anomaly
    df['anomaly'].loc[df['anomaly'] == 1] = 0
    df['anomaly'].loc[df['anomaly'] == -1] = 2
    df['anomaly_class'] = df['anomaly']
    max_anomaly_score = df['score'].loc[df['anomaly_class'] == 2].max()
    medium_percentile = df['score'].quantile(0.24)
    df['anomaly_class'].loc[(df['score'] > max_anomaly_score) & (df['score'] <= medium_percentile)] = 1
    return df
```

Les valeurs réelles des mesures sont indiquées dans la ligne bleue et les points d'anomalie sont mis en surbrillance en points rouges. Dans le tableau, le fond rouge indique des anomalies élevées et le jaune des anomalies faibles.

```
import warnings

test=metrics_df.iloc[:, [1,3,4]]

warnings.filterwarnings('ignore')

clf.fit(metrics_df.iloc[:, [1,3,4]])

pred = clf.predict(metrics_df.iloc[:,[1,3,4]])

test_df = pd.DataFrame()

test_df['Date & Time'] = metrics_df['Date & Time']

print(test_df)

# Find decision function to find the score and classify anomalies

test_df['score'] = clf.decision_function(metrics_df.iloc[:,[1,3,4]])

test_df['actuals'] = metrics_df.iloc[:, 1]

test_df['anomaly'] = pred

# Get the indexes of outliers in order to compare the metrics with use case anomalies if required

outliers = test_df.loc[test_df['anomaly'] == -1]

outlier_index = list(outliers.index)

test_df = classify_anomalies(test_df, metrics_df.columns[1])

print(metrics_df.columns[1])

print(test_df)

output_table_1=test_df.copy()

output_table_2 = metrics_df.copy()
```



(a) outliers_fraction = 0.01

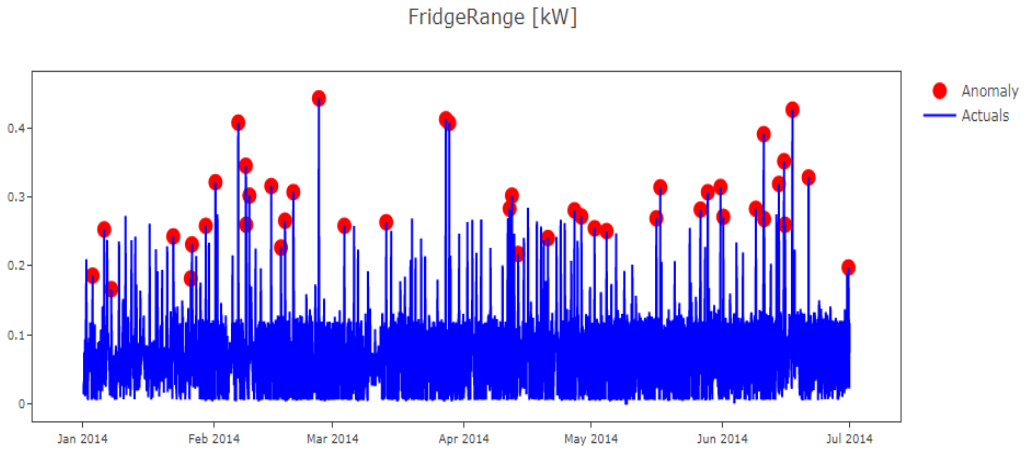


(b) outliers_fraction = 0.1

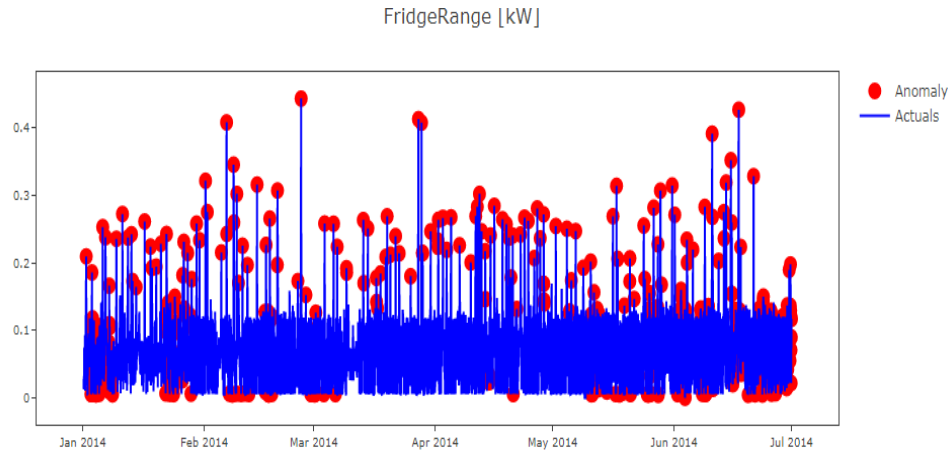
Date	Actual Values	% Change
2014063023	0.062	70.487
2014063022	0.018	1.868
2014063021	0.018	-3327.697
2014063020	0.612	-0.34
2014063019	0.614	1.281
2014063018	0.606	-1.603
2014063017	0.616	96.432

Date	Actual Values	% Change
2014063023	0.062	70.487
2014063022	0.018	1.868
2014063021	0.018	-3327.697
2014063020	0.612	-0.34
2014063019	0.614	1.281
2014063018	0.606	-1.603

Figure 14: Détection d'anomalies dans Furnace HRV avec deux paramètres (Isolation Forest)



(a) outliers_fraction = 0.01



(b) outliers_fraction = 0.1

Date	Actual Values	% Change
2014063023	0.117	23.403
2014063022	0.09	75.755
2014063021	0.022	-225.924
2014063020	0.071	-72.55
2014063019	0.122	-5.609
2014063018	0.129	-53.233
2014063017	0.198	31.133

Date	Actual Values	% Change
2014063023	0.117	23.403
2014063022	0.09	75.755
2014063021	0.022	-225.924
2014063020	0.071	-72.55
2014063019	0.122	-5.609
2014063018	0.129	-53.233

Figure 15: Détection d'anomalies dans FridgeRange avec deux paramètres (Isolation Forest)

La table conditionnelle nous donne des informations sur des cas tels que des données non présentes (la valeur est zéro) capturées en tant qu'anomalie élevée, ce qui pourrait être le résultat d'un pipeline interrompu dans le traitement des données, qu'il convient de corriger, ainsi que de mettre en évidence les anomalies de haut et de bas niveau.

Une amélioration serait de combiner un comportement anormal qui se produit continuellement. Par exemple, les grosses journées de charge qui entraîneraient un pic des statistiques pendant quelques jours pourraient être présentés comme un comportement unique.

IV.3.2 Méthode 2 : One Class SVM

Un autre algorithme d'apprentissage automatique populaire est la machine à vecteurs de support (SVM). De par sa conception, un SVM fonctionne comme un classificateur linéaire binaire, ce qui signifie qu'il ne peut classer que des échantillons dans une catégorie ou une autre. Pour ce faire, il sépare au mieux les échantillons d'apprentissage avec une limite linéaire, la meilleure frontière disposant de la plus grande marge entre les échantillons. Cependant, le SVM peut être utilisé lorsque les échantillons ne sont pas séparables de manière linéaire (par exemple, les données de consommation d'électricité) en implémentant une fonction de noyau.

Le but du SVM est alors de minimiser les erreurs pour tous les échantillons, tout en maintenant la marge entre les échantillons et la limite aussi grande que possible.

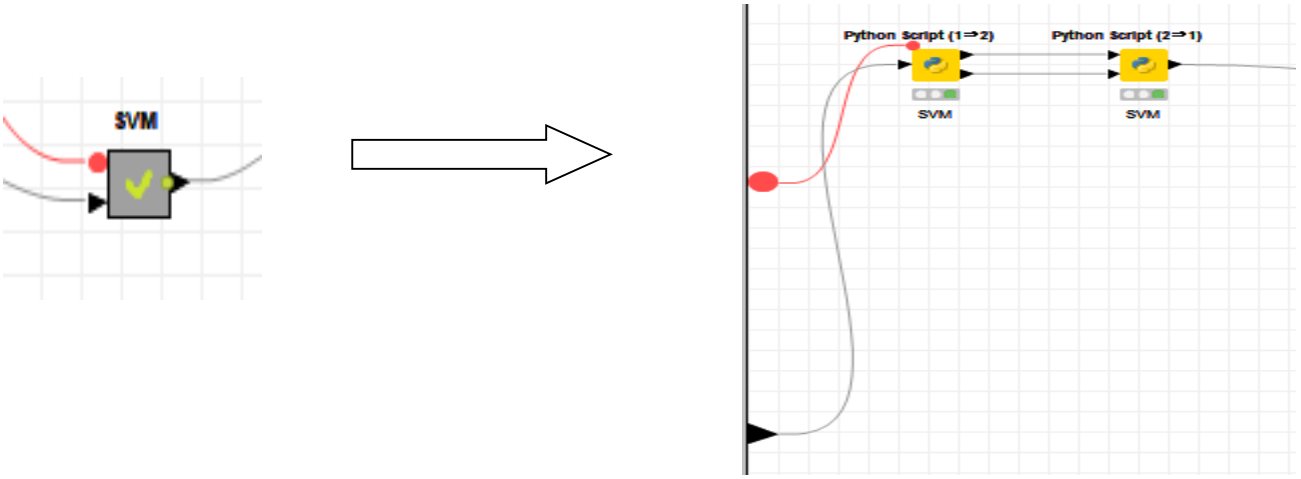
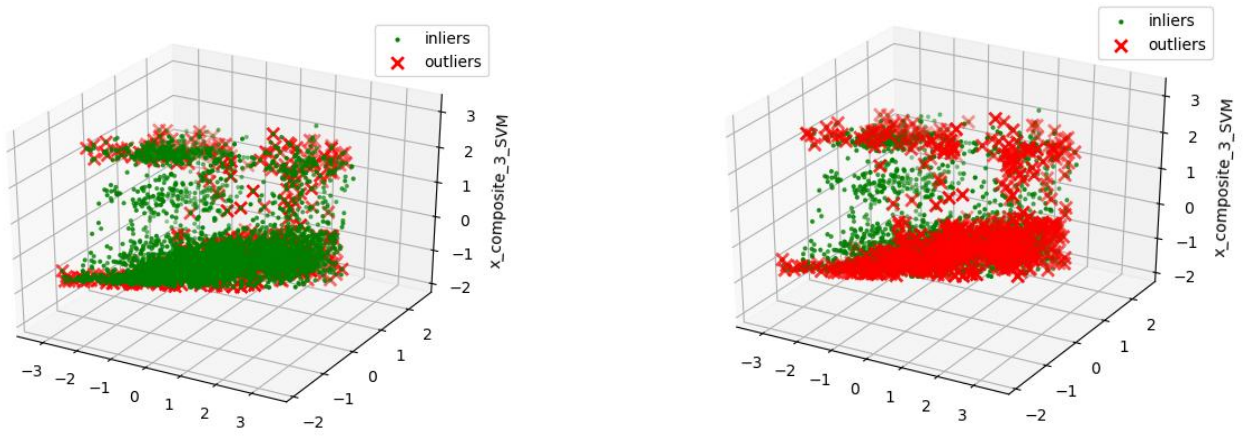


Figure 16: La méthode One-Class SVM

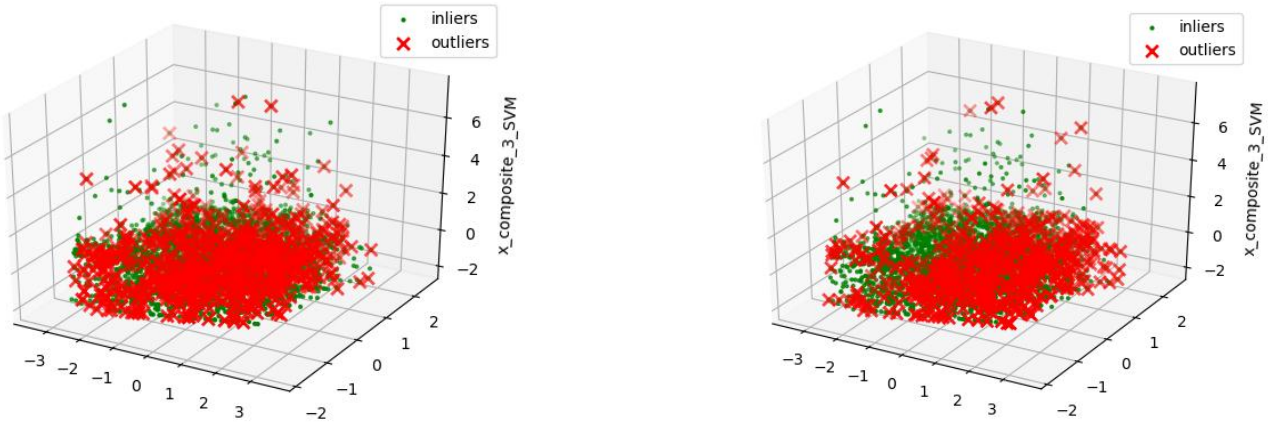
Le programme sous-jacent à cette méthode est :

```
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from mpl_toolkits.mplot3d import Axes3D
pca = PCA(n_components=3)  # Reduce to k=3 dimensions
scaler = StandardScaler()
# normalize the metrics
X = scaler.fit_transform(metrics_df[to_model_columns])
X_reduce = pca.fit_transform(X)
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.set_zlabel("x_composite_3_SVM")
# Plot the compressed data points
ax.scatter(X_reduce[:, 0], X_reduce[:, 1], zs=X_reduce[:, 2], s=4, lw=1, label="inliers", c="green")
# Plot x's for the ground truth outliers
ax.scatter(X_reduce[outlier_index, 0], X_reduce[outlier_index, 1], X_reduce[outlier_index, 2],
        lw=2, s=60, marker="x", c="red", label="outliers")
ax.legend()
plt.show()
```



(a) outliers_fraction = 0.01 (b) outliers_fraction = 0.1

Figure 17: Représentation 3D de Furnace HRV [kw] (1-Class SVM)



(a) outliers_fraction = 0.01 (b) outliers_fraction = 0.1

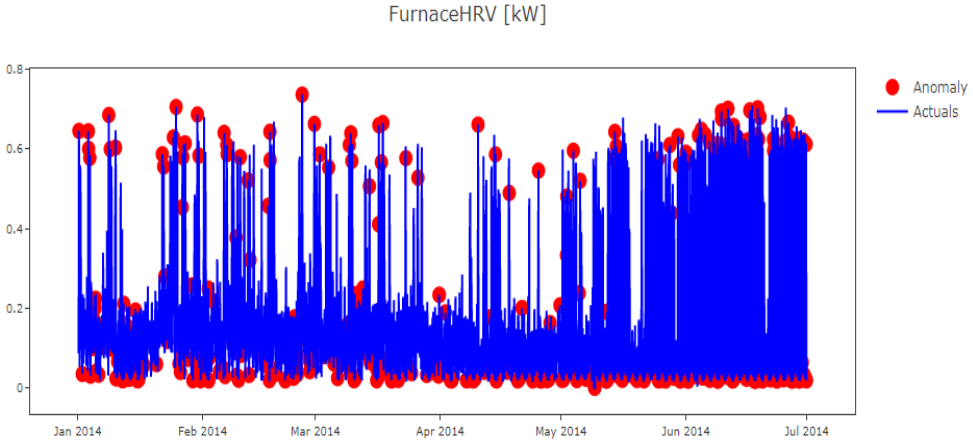
Figure 18: Représentation 3D de Fridge Range (1-Class SVM)

Nous identifions les anomalies pour les métriques individuelles et nous traçons les résultats.

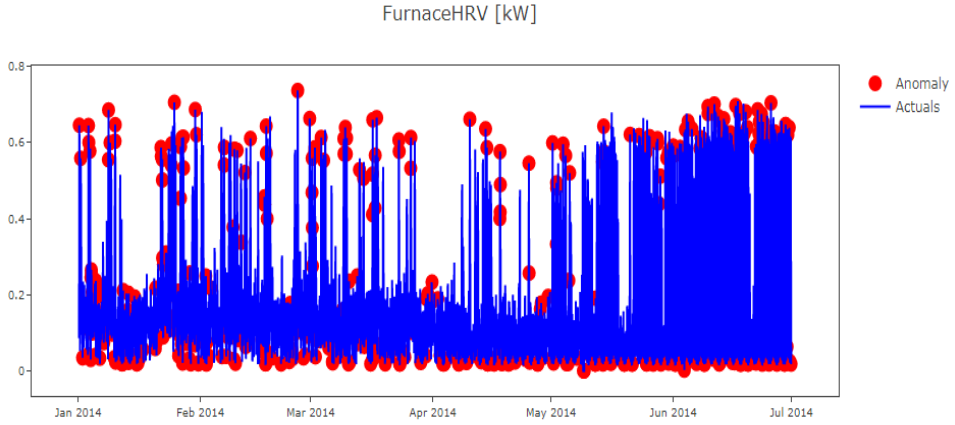
- Axe X – date
- Axe Y - Valeurs réelles et points d'anomalie.

Le programme sous-jacent à la méthode One-Class SVM est donné comme suit :

```
import warnings

test=metrics_df.iloc[:, [1,3,4]]

warnings.filterwarnings('ignore')

clf.fit(metrics_df.iloc[:, [1,3,4]])

pred = clf.predict(metrics_df.iloc[:,[1,3,4]])

test_df = pd.DataFrame()

test_df['Date & Time'] = metrics_df['Date & Time']

print(test_df)

# Find decision function to find the score and classify anomalies

test_df['score'] = clf.decision_function(metrics_df.iloc[:,[1,3,4]])

test_df['actuals'] = metrics_df.iloc[:, 1]

test_df['anomaly'] = pred

# Get the indexes of outliers in order to compare the metrics     with use case anomalies if required

outliers = test_df.loc[test_df['anomaly'] == -1]

outlier_index = list(outliers.index)

test_df = classify_anomalies(test_df, metrics_df.columns[1])

print(metrics_df.columns[1])

print(test_df)

output_table_1=test_df.copy()

output_table_2 = metrics_df.copy()
```



(a) outliers_fraction = 0.01

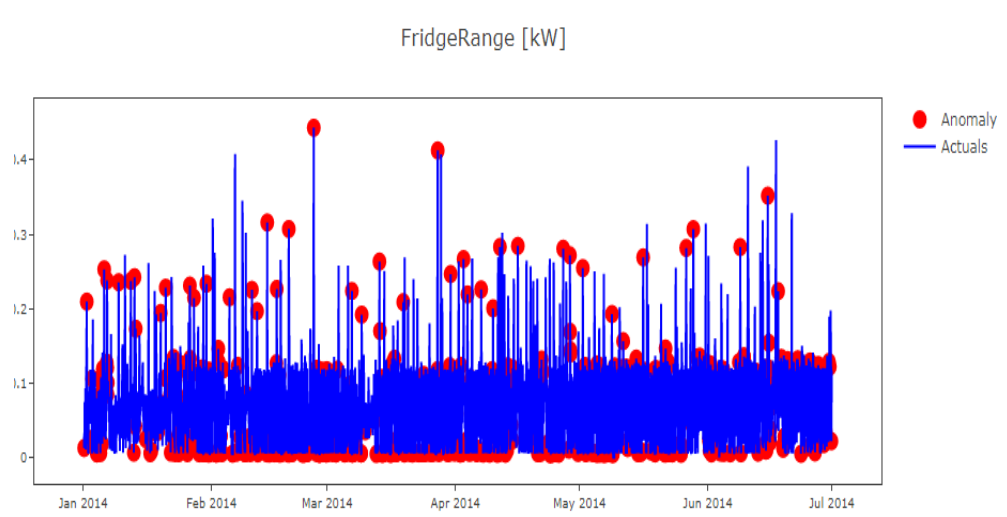


(b) outliers_fraction = 0.1

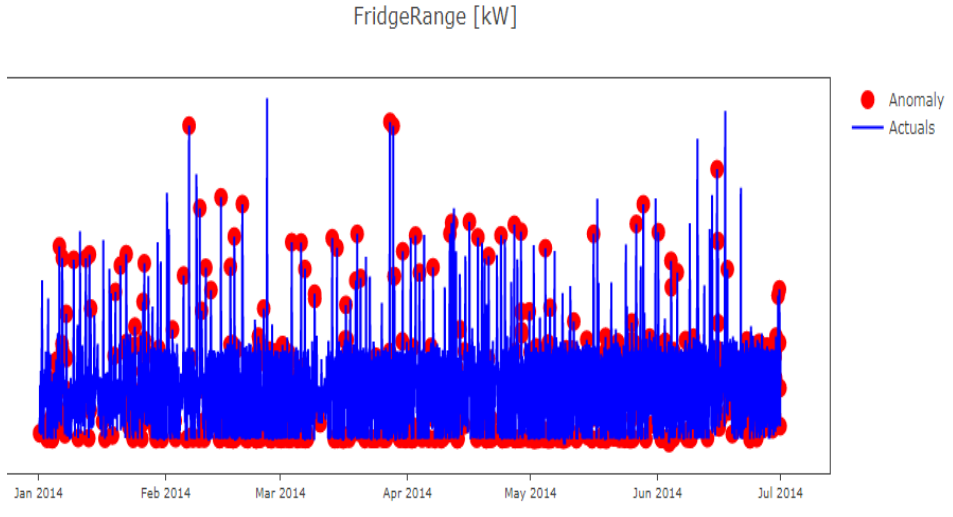
Date	Actual Values	% Change
2014063023	0.062	70.487
2014063022	0.018	1.868
2014063021	0.018	-3327.697
2014063020	0.612	-0.34
2014063019	0.614	1.281
2014063018	0.606	-1.603

Date	Actual Values	% Change
2014063023	0.062	70.487
2014063022	0.018	1.868
2014063021	0.018	-3327.697
2014063020	0.612	-0.34
2014063019	0.614	1.281
2014063018	0.606	-1.603

Figure 19: Détection d'anomalies dans Furnace HRV avec deux paramètres (1-Class SVM)



(a) outliers_fraction = 0.01



(b) outliers_fraction = 0.1

Date	Actual Values	% Change
2014063023	0.117	23.403
2014063022	0.09	75.755
2014063021	0.022	-225.924
2014063020	0.071	-72.55
2014063019	0.122	-5.609
2014063018	0.129	-53.233

Date	Actual Values	% Change
2014063023	0.117	23.403
2014063022	0.09	75.755
2014063021	0.022	-225.924
2014063020	0.071	-72.55
2014063019	0.122	-5.609
2014063018	0.129	-53.233

Figure 20: Détection d'anomalies dans Fridge Range avec deux paramètres (1-Class SVM)

IV.3.3 Méthode 3 : K-Means

K-Means est un algorithme de classification largement utilisé. Il crée «k» des groupes similaires de points de données. Les instances de données qui ne font pas partie de ces groupes peuvent potentiellement être marquées comme des anomalies.

L'hypothèse de base de la détection d'anomalie basée sur le regroupement est que, si nous regroupons les données, les données normales appartiendront à des clusters tandis que les anomalies n'appartiendront à aucun cluster ou à de petits clusters.

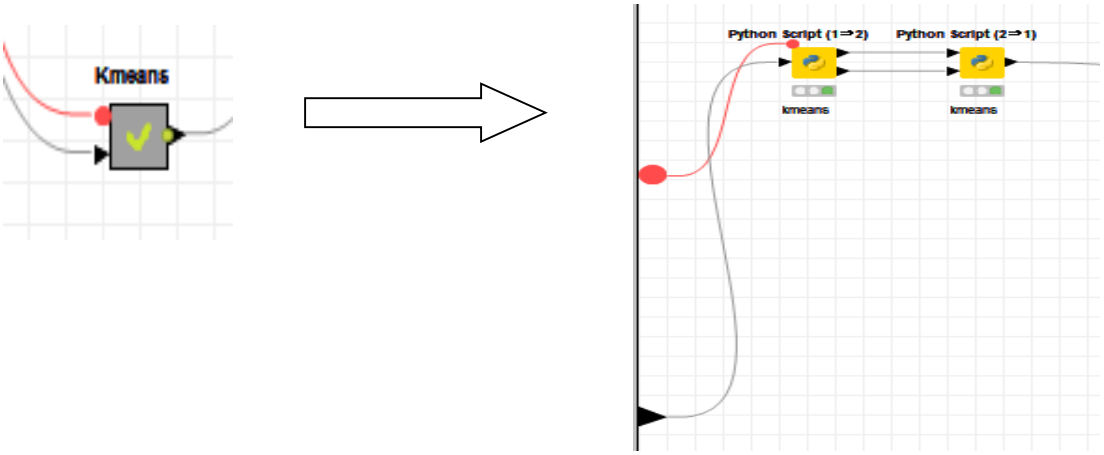


Figure 21: La méthode K-Means

Sur la courbe au coude ci-dessus, nous voyons que le graphique se stabilise après 8 clusters, ce qui implique que l'ajout davantage de grappes n'explique pas beaucoup plus la variance de notre variable pertinente. Nous définissons n_clusters = 8, et lors de la génération de la sortie K-Means, nous utilisons les données pour tracer les clusters 3D.

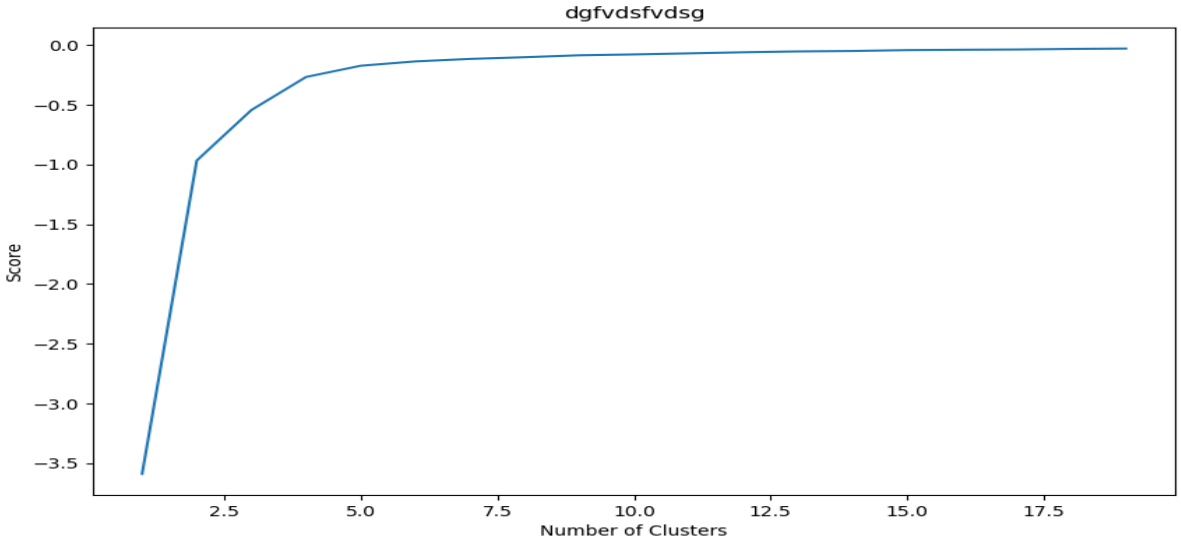
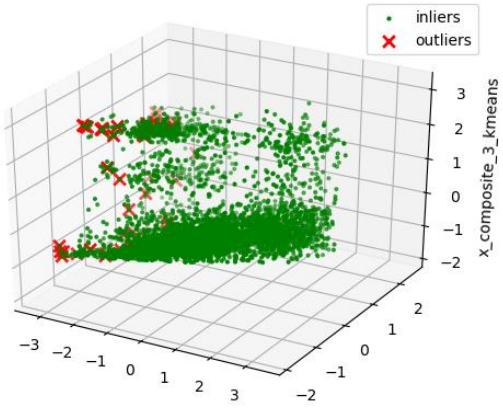


Figure 22: Nombre de clusters

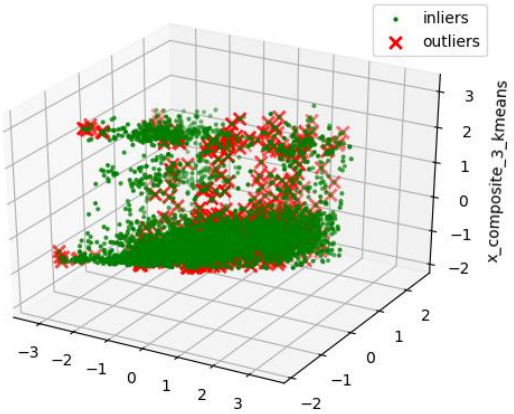
L'hypothèse de base de la détection d'anomalie basée sur le regroupement est que, si nous regroupons les données, les données normales appartiendront à des clusters tandis que les anomalies n'appartiendront à aucun cluster ou à de petits clusters. Nous utilisons les étapes suivantes pour rechercher et visualiser les anomalies. Nous calculons la distance entre chaque point et son centroid le plus proche. Les plus grandes distances sont considérées comme des anomalies. Puis nous définissons le seuil comme distance minimale de ces valeurs éloignées.

Le programme suivant représente le contexte d'exécution de la méthode K-Means.

```
# get the distance between each point and its nearest centroid.
getDistanceByPoint(metrics_df[to_model_columns], clf)
number_of_outliers = int(outliers_fraction*len(distance))
threshold = distance.nlargest(number_of_outliers).min()
# anomaly1 contain the anomaly result of the above method Cluster (0:normal, 1:anomaly)
metrics_df['anomaly'] = (distance >= threshold).astype(int)
metrics_df['anomaly'].loc[metrics_df['anomaly'] == 1] = -1
metrics_df['anomaly'].loc[metrics_df['anomaly'] == 0] = 1
outliers = metrics_df.loc[metrics_df['anomaly'] == -1]
outlier_index = list(outliers.index)
pca = PCA(n_components=3)  # Reduce to k=3 dimensions
scaler = StandardScaler()
# normalize the metrics
X = scaler.fit_transform(metrics_df[to_model_columns])
X_reduce = pca.fit_transform(X)
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.set_zlabel("x_composite_3_kmeans")
# Plot the compressed data points
ax.scatter(X_reduce[:, 0], X_reduce[:, 1], zs=X_reduce[:, 2], s=4, lw=1, label="inliers", c="green")
ax.scatter(X_reduce[outlier_index, 0], X_reduce[outlier_index, 1], X_reduce[outlier_index, 2],
        lw=2, s=60, marker="x", c="red", label="outliers")
ax.legend()
plt.show()
```

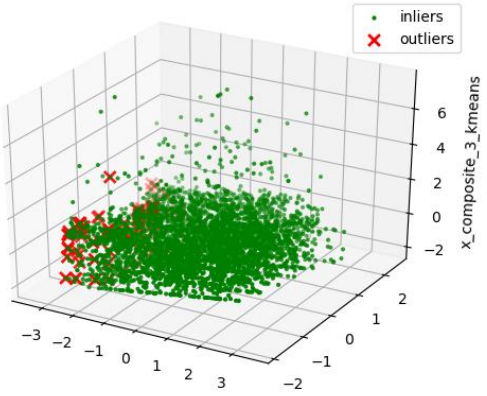


(a) outliers_fraction = 0.01

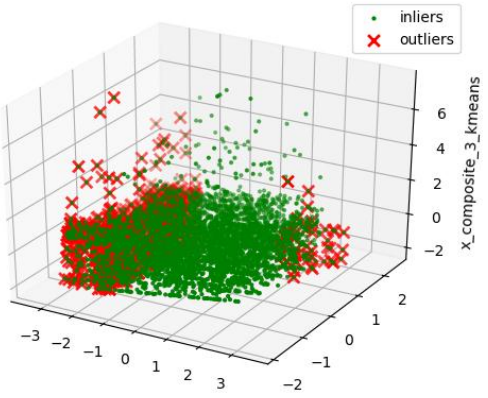


(b) outliers_fraction = 0.1

Figure 23: Représentation 3D de Furnace HRV [kw] (K-Means)



(a) outliers_fraction = 0.01



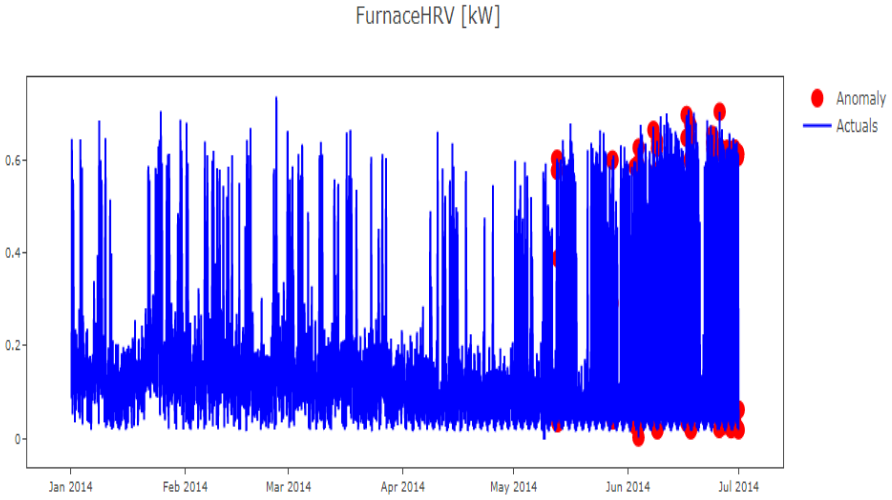
(b) outliers_fraction = 0.1

Figure 24 : Représentation 3D de Fridge Range (K-Means)

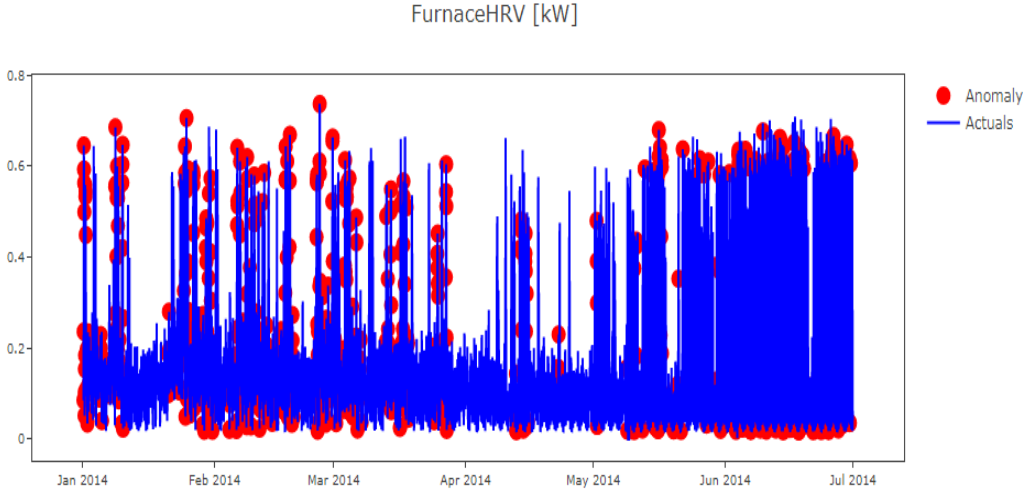
Visualisation des résultats :

```
def classify_anomalies(df, metric_name):
    df['metric_name'] = metric_name
    df = df.sort_values(by='Date & Time', ascending=False)
    # Shift actuals by one timestamp to find the percentage chage between current and previous data point
    df['shift'] = df['actuals'].shift(-1)
    df['percentage_change'] = ((df['actuals'] - df['shift']) / df['actuals']) * 100
    # Categorise anomalies as 0-no anomaly, 1- low anomaly , 2 - high anomaly
    df['anomaly'].loc[df['anomaly'] == 1] = 0
    df['anomaly'].loc[df['anomaly'] == -1] = 2
    df['anomaly_class'] = df['anomaly']
    max_anomaly_score = df['score'].loc[df['anomaly_class'] == 2].max()
    medium_percentile = df['score'].quantile(0.24)
    df['anomaly_class'].loc[(df['score'] > max_anomaly_score) & (df['score'] <= medium_percentile)] = 1
     return df
```

```
import warnings
test=metrics_df.iloc[:, [1,3,4]]
warnings.filterwarnings('ignore')
clf.fit(metrics_df.iloc[:, [1,3,4]])
pred = clf.predict(metrics_df.iloc[:,[1,3,4]])
test_df = pd.DataFrame()
test_df['Date & Time'] = metrics_df['Date & Time']
print(test_df)
# Find decision function to find the score and classify anomalies
test_df['score'] = getDistanceByPoint(metrics_df.iloc[:,[1,3,4]], clf)
test_df['actuals'] = metrics_df.iloc[:, 1]
distance = getDistanceByPoint(metrics_df.iloc[:,[1,3,4]], clf)
number_of_outliers = int(outliers_fraction*len(distance))
threshold = distance.nlargest(number_of_outliers).min()
# anomaly1 contain the anomaly result of the above method Cluster (0:normal, 1:anomaly)
test_df['anomaly'] = (distance >= threshold).astype(int)
test_df['anomaly'].loc[test_df['anomaly'] == 1] = -1
test_df['anomaly'].loc[test_df['anomaly'] == 0] = 1
# Get the indexes of outliers in order to compare the metrics     with use case anomalies if required
outliers = test_df.loc[test_df['anomaly'] == -1]
outlier_index = list(outliers.index)
test_df = classify_anomalies(test_df, metrics_df.columns[1])
print(metrics_df.columns[1])
print(test_df)
output_table_1=test_df.copy()
output_table_2 = metrics_df.copy()
```



(a) outliers_fraction=0.01

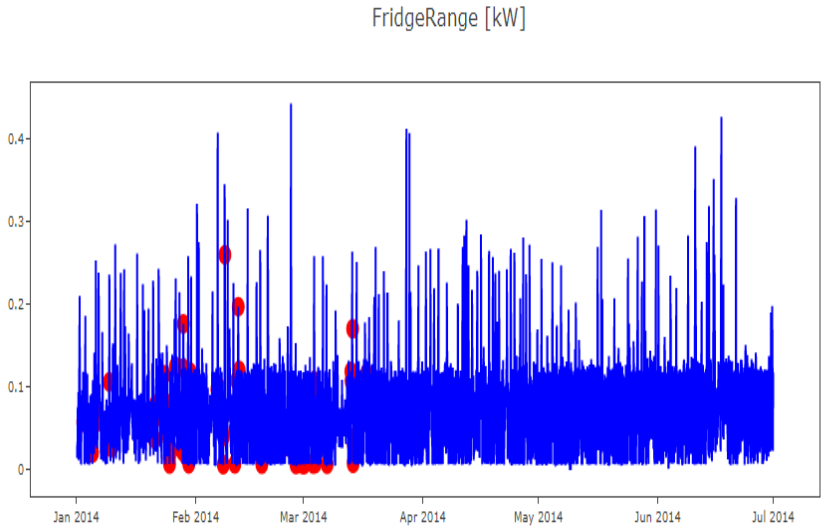


(b) outliers_fraction=0.1

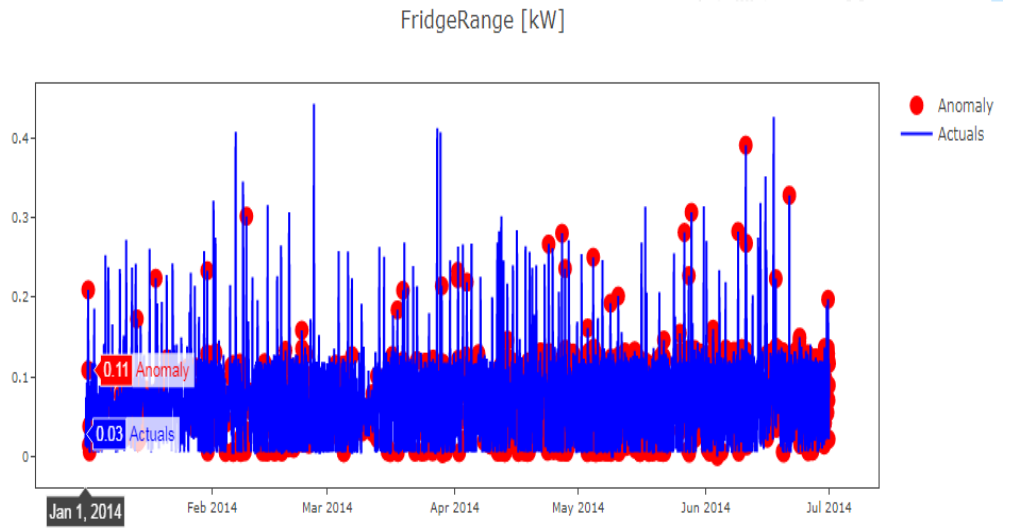
Date	Actual Values	% Change
2014063023	0.062	70.487
2014063022	0.018	1.868
2014063021	0.018	-3327.697
2014063020	0.612	-0.34
2014063019	0.614	1.281
2014063018	0.606	-1.603

Date	Actual Values	% Change
2014063023	0.062	70.487
2014063022	0.018	1.868
2014063021	0.018	-3327.697
2014063020	0.612	-0.34
2014063019	0.614	1.281
2014063018	0.606	-1.603

Figure 25: Détection d'anomalies dans Furnace HRV avec deux paramètres (K-Means)



(a) outliers_fraction=0.01



(b) outliers_fraction=0.1

Date	Actual Values	% Change
2014063023	0.117	23.403
2014063022	0.09	75.755
2014063021	0.022	-225.924
2014063020	0.071	-72.55
2014063019	0.122	-5.609
2014063018	0.129	-53.233

Date	Actual Values	% Change
2014063023	0.117	23.403
2014063022	0.09	75.755
2014063021	0.022	-225.924
2014063020	0.071	-72.55
2014063019	0.122	-5.609
2014063018	0.129	-53.233

Figure 26: Détection d'anomalies dans Fridge Range avec deux paramètres (K-Means)

IV.4 L'évaluation et analyse des résultats

Les résultats obtenus sont ensuite exploités, et nous cherchons à les analyser (leur donner un sens).

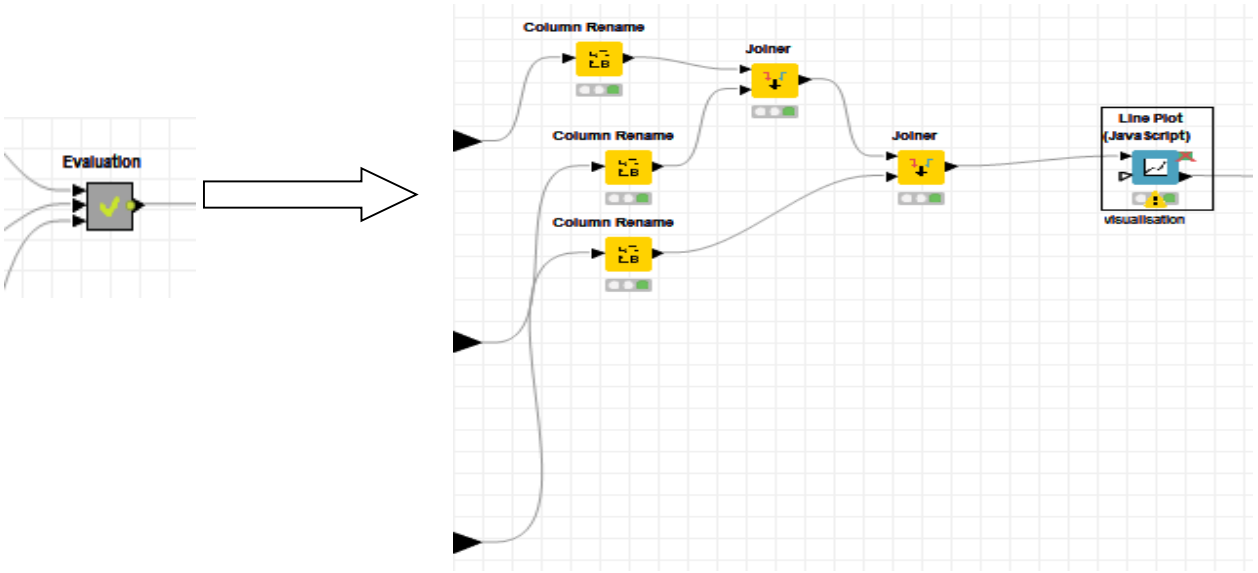


Figure 27: Exploitation des données

Dans cette étape, on collecte les résultats d'anomalie pour chaque méthode et on l'introduisant dans le nœud de visualisation pour avoir la table finale et le graphe associé.

IV.4.1 Exploitation des résultats

Finalement les résultats de l'analyse permettent de répondre aux "problèmes posés" et contribuent donc à l'aide à la décision.

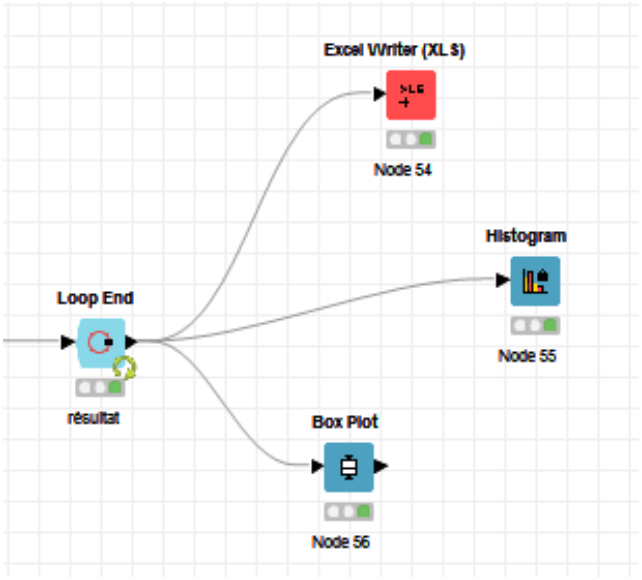


Figure 28: Affichage des résultats

Le nœud LOOP END est un nœud de fin d'une boucle. Il est utilisé pour marquer la fin d'une boucle de flux de travail et collecte les résultats intermédiaires par concaténation par rangée des tables entrantes.

Tableau 7: Résultats avec outlier_fraction = 0.01

Date & Tir	actuals	anomaly_is	metric_name	shift_isol	percentage_	anomaly_class_is	anomaly_SVM	shift_SVM	percentag	anomaly_class_SV	anomaly_KM	shift_KMe	percentag	anomaly_class	Iteration
201406302	0,061675	0	FurnaceHRV [I	0,018202	70,4868715	1	0	0,018202	70,48687	0	2	0,018202	70,48687	2	0
201406302	0,018202	0	FurnaceHRV [I	0,017862	1,86790382	1	2	0,017862	1,867904	2	2	0,017862	1,867904	2	0
201406302	0,017862	0	FurnaceHRV [I	0,612263	-3327,69661	1	0	0,612263	-3327,7	1	2	0,612263	-3327,7	2	0
201406302	0,612263	2	FurnaceHRV [I	0,614347	-0,3403586	2	2	0,614347	-0,34036	2	2	0,614347	-0,34036	2	0
201406301	0,614347	2	FurnaceHRV [I	0,606479	1,280583655	2	0	0,606479	1,280584	1	2	0,606479	1,280584	2	0
201406301	0,606479	2	FurnaceHRV [I	0,616203	-1,60333365	2	0	0,616203	-1,60333	1	2	0,616203	-1,60333	2	0
201406301	0,616203	2	FurnaceHRV [I	0,021988	96,43164329	2	0	0,021988	96,43164	0	2	0,021988	96,43164	2	0
201406301	0,021988	0	FurnaceHRV [I	0,022982	-4,51755028	1	2	0,022982	-4,51755	2	2	0,022982	-4,51755	2	0
201406301	0,022982	0	FurnaceHRV [I	0,024286	-5,67360497	1	0	0,024286	-5,6736	1	0	0,024286	-5,6736	0	0
201406301	0,024286	0	FurnaceHRV [I	0,025206	-3,78825998	1	2	0,025206	-3,78826	2	0	0,025206	-3,78826	0	0
201406301	0,025206	0	FurnaceHRV [I	0,606091	-2304,5911	1	0	0,606091	-2304,59	0	0	0,606091	-2304,59	0	0
201406301	0,606091	0	FurnaceHRV [I	0,608065	-0,32576716	1	0	0,608065	-0,32577	0	0	0,608065	-0,32577	0	0
201406301	0,608065	0	FurnaceHRV [I	0,639359	-5,14656229	1	0	0,639359	-5,14656	1	0	0,639359	-5,14656	0	0
201406301	0,639359	0	FurnaceHRV [I	0,609042	4,741898674	1	0	0,609042	4,741899	1	0	0,609042	4,741899	0	0

Tableau 8: Résultats avec outlier_fraction = 0.1

Date & Tir	actuals	anomaly_is	metric_nan	shift_isol	percentag	anomaly_class_is	anomaly_SVM	shift_SVM	percentag	anomaly_	anomaly_	shift_KMe	percentag	anomaly_class_KM	Iteration
201406302	0,061675	2	FurnaceHR\	0,018202	70,48687	2	0	0,018202	70,48687	0	0	0,018202	70,48687	0	0
201406302	0,018202	2	FurnaceHR\	0,017862	1,867904	2	0	0,017862	1,867904	1	0	0,017862	1,867904	0	0
201406302	0,017862	2	FurnaceHR\	0,612263	-3327,7	2	2	0,612263	-3327,7	2	0	0,612263	-3327,7	0	0
201406302	0,612263	2	FurnaceHR\	0,614347	-0,34036	2	0	0,614347	-0,34036	1	0	0,614347	-0,34036	0	0
201406301	0,614347	2	FurnaceHR\	0,606479	1,280584	2	0	0,606479	1,280584	0	0	0,606479	1,280584	0	0
201406301	0,606479	2	FurnaceHR\	0,616203	-1,60333	2	0	0,616203	-1,60333	1	0	0,616203	-1,60333	0	0
201406301	0,616203	2	FurnaceHR\	0,021988	96,43164	2	0	0,021988	96,43164	0	0	0,021988	96,43164	0	0
201406301	0,021988	2	FurnaceHR\	0,022982	-4,51755	2	0	0,022982	-4,51755	1	0	0,022982	-4,51755	0	0
201406301	0,022982	0	FurnaceHR\	0,024286	-5,6736	1	0	0,024286	-5,6736	1	0	0,024286	-5,6736	0	0
201406301	0,024286	0	FurnaceHR\	0,025206	-3,78826	1	2	0,025206	-3,78826	2	0	0,025206	-3,78826	0	0
201406301	0,025206	0	FurnaceHR\	0,606091	-2304,59	0	0	0,606091	-2304,59	1	0	0,606091	-2304,59	0	0
201406301	0,606091	0	FurnaceHR\	0,608065	-0,32577	1	0	0,608065	-0,32577	0	2	0,608065	-0,32577	2	0
201406301	0,608065	0	FurnaceHR\	0,639359	-5,14656	1	0	0,639359	-5,14656	1	2	0,639359	-5,14656	2	0
201406301	0,639359	2	FurnaceHR\	0,609042	4,741899	2	2	0,609042	4,741899	2	0	0,609042	4,741899	0	0
201406300	0,609042	0	FurnaceHR\	0,032277	94,70033	1	0	0,032277	94,70033	0	0	0,032277	94,70033	0	0
201406300	0,032277	0	FurnaceHR\	0,092592	-186,864	0	0	0,092592	-186,864	1	0	0,092592	-186,864	0	0
201406300	0,092592	0	FurnaceHR\	0,035432	61,73282	0	0	0,035432	61,73282	0	0	0,035432	61,73282	0	0
201406300	0,035432	0	FurnaceHR\	0,036255	-2,32212	0	0	0,036255	-2,32212	0	2	0,036255	-2,32212	2	0
201406300	0,036255	0	FurnaceHR\	0,617523	-1603,28	0	0	0,617523	-1603,28	0	2	0,617523	-1603,28	2	0
201406300	0,617523	2	FurnaceHR\	0,619959	-0,39441	2	0	0,619959	-0,39441	1	2	0,619959	-0,39441	2	0

Le nœud Histogramme permet de visualiser les anomalies ce qui nous aide à mieux comparer les performances des trois méthodes.

Figure 29: Comparaison des anomalies des 3 méthodes (outlier_fraction = 0.01)

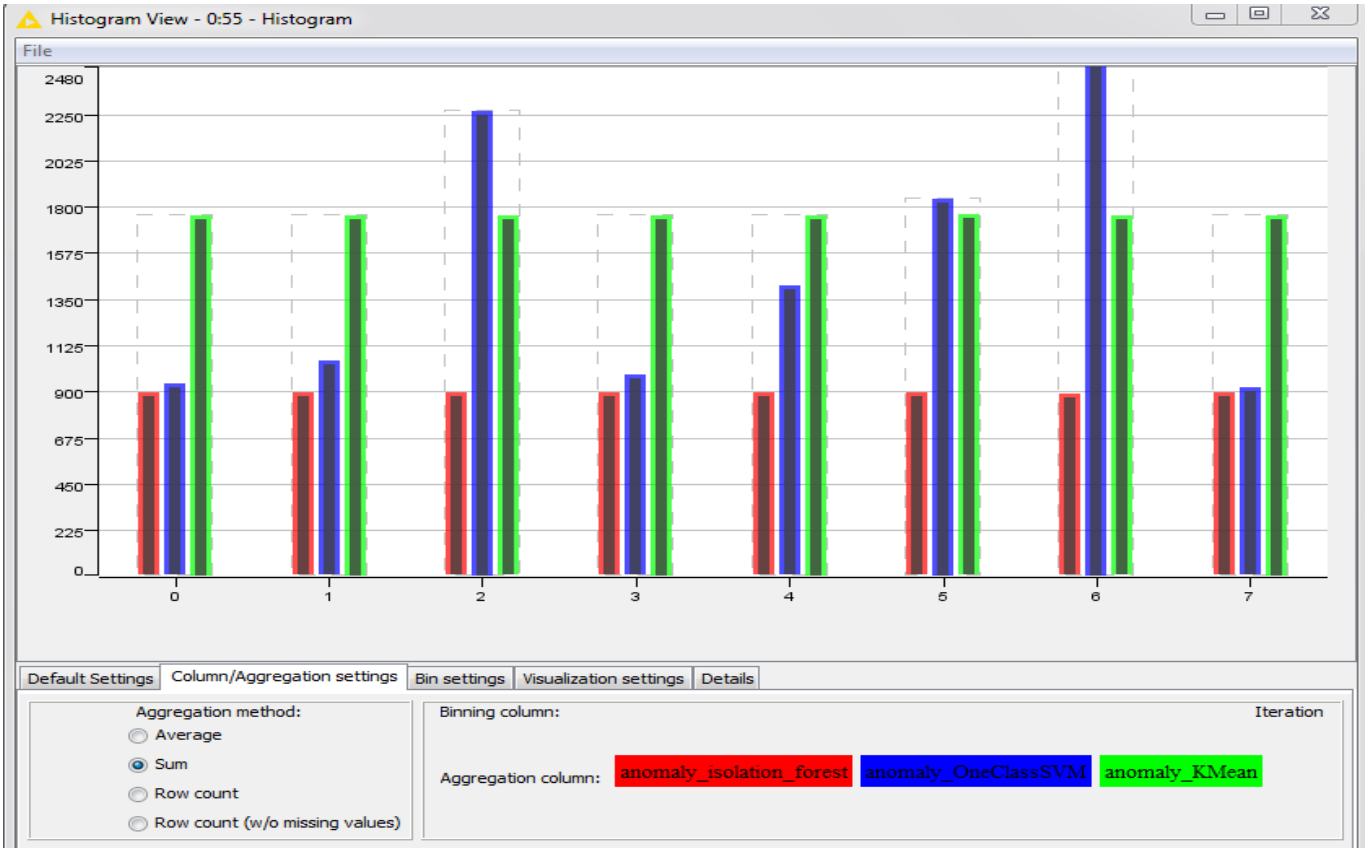


Figure 30: Comparaison des anomalies des 3 méthodes (outlier_fraction = 0.1)

Nous remarquons que le modèle qui donne plus de précision est le modèle SVM. Cet exemple fait ressortir la complémentarité entre l'analyse descriptive et l'analyse prédictive.

IV.4.2 Comparaison

Le tableau ci-dessous présente les résultats pour les trois méthodes que nous avons implémentées. Les mesures d'erreur décrivent le même résultat mais pour les données de ce projet, SVM est le plus précis. L'erreur est mesurée en utilisant un flux de données (01/01/2014 au 30/06/2014). D'après l'étude l'algorithme de 1-Class SVM est le meilleur modèle.

Tableau 9 : Mesures d'erreur par les trois méthodes (outlier_fraction=0.01)

row ID	score_isolation_forest	actuals	anomaly_isolation_forest	shift_isolation_forest	anomaly_class_isolation_forest	score_OneClassSVM	anomaly_OneClassSVM	shift_OneClassSVM	anomaly_class_OneClassSVM	score_KMean	anomaly_KMean	shift_KMean	anomaly_class_KMean	Iteration
Minimum	-0,224802117	-13,72	0	-13,72	0	-0,00049761	0	-13,72	0	4,80273669	0	-13,72	0	0
Smallest	-0,103681842	-0,1	0	-0,1	0	-0,00049761	0	-0,1	0	4,80273669	0	-0,1	0	0
Lower Quartile	-0,01201712	0,003343333	0	0,003343611	0	0,00013193	0	0,003343611	0	26,4629836	0	0,003343611	0	1,5
Median	0,018934143	0,021516945	0	0,021559167	0	0,001012059	0	0,021559167	0	38,058315	0	0,021559167	0	3,5
Upper Quartile	0,049105666	0,116044722	0	0,116081667	0	0,018887806	0	0,116081667	0	50,0881201	0	0,116081667	0	5,5
Largest	0,096840492	0,284396111	0	0,284396111	0	0,047020464	0	0,284396111	0	85,4151515	0	0,284396111	0	7
Maximum	0,096840492	85,48	2	#NOMBRE!	2	0,08255756	2	#NOMBRE!	2	101,037144	2	#NOMBRE!	2	7

Tableau 10: Mesures d'erreur par les trois méthodes (outlier_fraction=0.1)

row ID	score_isolation_forest	actuals	anomaly_isolation_forest	shift_isolation_forest	anomaly_class_isolation_forest	score_OneClassSVM	anomaly_OneClassSVM	shift_OneClassSVM	anomaly_class_OneClassSVM	score_KMean	anomaly_KMean	shift_KMean	anomaly_class_KMean	Iteration
Minimum	-0,228515132	-13,72	0	-13,72	0	-3,36673404	0	-13,72	0	4,65162754	0	-13,72	0	0
Smallest	-0,101032121	-0,1	0	-0,1	0	-0,28808018	0	-0,1	0	4,65162754	0	-0,1	0	0
Lower Quartile	-0,013240401	0,003343333	0	0,003343611	0	0,000175826	0	0,00334361	0	24,5773341	0	0,003343611	0	1,5
Median	0,017091193	0,021516945	0	0,021559167	0	0,00524431	0	0,02155917	0	36,8746794	0	0,021559167	0	3,5
Upper Quartile	0,045301111	0,116044722	0	0,116081667	0	0,193258072	0	0,11608167	0	49,6356758	0	0,116081667	0	5,5
Largest	0,099776847	0,284396111	0	0,284396111	0	0,482693908	0	0,28439611	0	87,2159523	0	0,284396111	0	7
Maximum	0,099776847	85,48	2	#NOMBRE!	2	0,791227487	2	#NOMBRE!	2	93,6673073	2	#NOMBRE!	2	7

Dans ce travail, les anomalies ont été détectées avec trois méthodes pour chaque capteur placé au niveau d'un équipement. Il s'agit d'une comparaison entre trois méthodes pour illustrer celle qui permettra la détection des anomalies avec une grande précision.

Dans les tableaux contenant les résultats, les anomalies des différents capteurs et pour chaque méthode nous a donné presque les mêmes résultats voir le pourcentage de change de chaque méthode est identique par contre les anomalies se diffère. On remarque que la méthode de 1-Class SVM détecte au mieux les anomalies.

IV.5 Conclusion

La détection d'anomalies est un problème complexe sans solution uniformément meilleure car le choix de la méthode à utiliser dépend largement du contexte, des propriétés des variables et données observées, ainsi que de l'objectif poursuivi. La recherche d'anomalies sur les données de

prévision de dépassement de seuil de zone montre, sur cet exemple, que chaque méthode projette sa conception de ce qu'est une anomalie.

En résumé, dans ce chapitre, trois méthodes ont été utilisées pour créer un modèle de prévision de la consommation d'électricité données : Isolation Forest, Support Vector Machine SVM et K-Means. Pour chaque méthode, on a assigné deux valeurs au paramètre outlier-fraction = 0.1 et 0.01 pour connaitre la méthode la plus sensible aux anomalies. En outre, pour chaque capteur plusieurs anomalies ont été détectées et affichées. Quelques anomalies majeures ont été expliquées et le reste inexpliqué des anomalies contiennent des données réelles de l'électricité gaspillée.

Conclusion générale

Conclusion générale

Le but de ce projet est d'utiliser des méthodes de détection des anomalies dans les séries temporelles pour identifier les défaillances électriques et l'énergie gaspillée dans les bâtiments résidentiels. Dans ce contexte, nous avons étudié les méthodes les plus répandues issues de l'apprentissage non supervisé et nous avons atteint une meilleure compréhension de différentes directions de recherche sur l'analyse des données en particulier celles qui sont particulières et qui peuvent refléter des anomalies.

Par ailleurs, la détection d'anomalies est un problème complexe sans solution uniformément meilleure car le choix de la méthode à utiliser dépend largement du contexte, des propriétés des variables et données observées, ainsi que de l'objectif poursuivi. La recherche d'anomalies sur les données de prévision montre, sur cet exemple, que chaque méthode projette sa conception de ce qu'est une anomalie.

Dans le cadre de ce projet de fin d'études, nous avons utilisées trois méthodes issues de l'apprentissage non supervisé pour la détection des anomalies dans la consommation d'électricité dans les bâtiments. Ces méthodes sont : Isolation Forest, One-Class SVM et K-Means.

Les résultats obtenus ont montré que chaque méthode estime une telle donnée comme anomalie ou non et que la méthode basée sur One-Class SVM a montré ses performances comparativement aux autres.

Perspectives :

- Implémenter d'autres méthodes d'apprentissage non supervisé
- Faire appel à un apprentissage supervisé et comparé les résultats avec ceux fournis par un apprentissage non supervisé en travaillant sur des bases de données réelles.

Références bibliographiques

Références bibliographiques

[1] Anthony Ferretti. ”Smart grids” : les réseaux et compteurs d’électricité intelligents : émergence d’une ère post-carbone, ou avènement de la société de contrôle. Art et histoire de l’art. 2014. dumas-0106418.

[2] BRAHIMI Karim, BOUCHALA Sofiane, ”Les aspects techniques des projets Smarts Grids”, Mémoire de Master ”Réseau électrique”, Faculté de Technologie, Université A/MIRA de Bejaia, 2013.

[3] "Smart grids (réseaux intelligents) : définition, applications et enjeux", https://www.connaissancedesenergies.org/fiche-pedagogique/reseau-intelligent-smart-grid, (consulté le 09 /05/2019).

[4] Sabrina Bouvier, Pascale Strubel, "Déployer un réseau plus intelligent grâce à des solutions et services de câblage", livre blanc, Mars 2010.

[5] Rim Badreddine. "Gestion énergétique optimisée pour un bâtiment intelligent multi-sources multi-charges : différents principes de validations". Rapport, Université de Grenoble, 2012.

[6] "La Gestion Technique de Bâtiment Pour des bâtiments intelligents", https://docplayer.fr/4225546-La-gestion-technique-de-batiment-pour-des-batiments-intelligents.html, (consulté le 05 /05/2019).

[7] Christian Gagné, "Introduction à l’apprentissage automatique", Support de cours, Apprentissage et reconnaissance, Université de Laval, Canada, Septembre 2016.

[8] Ali Yazid Ziat. "Apprentissage de représentation pour la prédiction et la classification de séries temporelles". Rapport sur les Réseaux de neurones [cs.NE]. Université Pierre et Marie Curie - Paris VI, 2017.

[9] Zakariyaa ISMAILI, Machine Learning, https://le-datascientist.fr/apprentissage-supervise-vs-non-supervise, (publié le 28 /01/2019).

[10] REGUIEG Samia et ZEGHABA Achouak. Étude comparative de quelques outils *de* classification sur des données médicales, Master Génie Biomédical, Université de Tlemcen, 2017.

[11] Chloé-Agathe Azencott, "Introduction au Machine Learning", Collection InfoSup, Dunod, Septembre 2018.

[12] Détection de valeurs aberrantes avec forêt d'isolation étendue – Vers la science des données, https://supportivy.com/detection-de-valeurs-aberrantes-avec-foret-disolation-etendue-vers-la-science-des-donnees/ (consulté le 11/03/2019).

[13] Ardjani Fatima, "Optimisation des SVM multi classes par des m´méthodes´évolutionnaires (PSO-SVM)", Mémoire de Magister, Université des Sciences et de la Technologie Mohamed Boudiaf,

[14] Mohamed BOUAZIZ, "Réseaux de neurones récurrents pour la classification de séquences dans des flux audiovisuels parallèles", Thèse de Doctorat, Université d'Avignon et des Pays de Vaucluse, décembre 2017.

[15] RANDRIANARIVONY Mamitiana Ignace, "Détection de concepts et annotation automatique d'images médicales par apprentissage profond", Master Université d'ANTANANARIVO et Toulouse, 2018.

[16] Best Python libraries for Machine Learning, https://www.geeksforgeeks.org/best-python-libraries-for-machine-learning/, (consulté 15/03/2019).

[17] Sean Barker, Aditya Mishra, David Irwin, Emmanuel Cecchet, Prashant Shenoy, and Jeannie Albrecht, ([Home dataset] Smart*: An Open Data Set and Tools for Enabling Research in Sustainable Homes, Proceedings of the 2012 Workshop on Data Mining Applications in Sustainability (SustKDD 2012), Beijing, China, August 2012).

[18] Sean Barker, Aditya Mishra, David Irwin, Prashant Shenoy, and Jeannie Albrecht, [Home dataset] SmartCap: Flattening Peak Electricity Demand in Smart Homes**.** *Proceedings of the 10th IEEE International Conference on Pervasive Computing and Communications (PerCom 2012)*, Lugano, Switzerland, March 2012.

Résumé: La détection des anomalies dans la consommation d'électricité en utilisant des méthodes de détection des outliers

Le but de ce projet est d'utiliser des méthodes de détection des anomalies dans les séries temporelles pour identifier les défaillances électriques et l'énergie gaspillée dans les bâtiments résidentiels. Dans ce contexte, nous avons utilisées trois méthodes issues de l'apprentissage non supervisé pour la détection des anomalies dans la consommation d'électricité dans les bâtiments. Ces méthodes sont : Isolation Forest, One-Class SVM et K-Means.

Les résultats obtenus ont montré que chaque méthode estime une telle donnée comme anomalie ou non et que la méthode basée sur One-Class SVM a montré ses performances comparativement aux autres.

Mots clés : Apprentissage non supervisé, les réseaux de capteurs sans fil, Isolation Forest, One-Class SVM, K-Means, KNIME.

Abstract: Detecting anomalies in electricity consumption by using outlier detection methods

The goal of this project is to use time series anomaly detection methods to identify electrical failures and wasted energy in residential buildings. In this context, we used three methods from unsupervised learning to detect anomalies in electricity consumption in buildings. These methods are: Forest Isolation, One-Class SVM, and K-Means.

The results obtained showed that each method estimates such a data as an anomaly or not and that the method based on One-Class SVM showed its performance compared to the others.

Keywords: Unsupervised Learning, Wireless Sensor Networks, Forest Isolation, One-Class SVM, K-Means, KNIME

ملخص: اكتشاف الحالات الشاذة في استهلاك الكهرباء باستخدام طرق الكشف عن الخارج

الهدف من هذا المشروع هو استخدام طرق الكشف عن الحالات الشاذة في السلاسل الزمنية لتحديد الأعطال الكهربائية والطاقة المهدرة في المباني السكنية .في هذا السياق ، استخدمنا ثلاث طرق من التعلم غير الخاضع للرقابة لاكتشاف الحالات الشاذة في استهلاك الكهرباء في المباني .هذه الطرق هي: عزل الغابات و SVM أحادية الطبقة و K-Means .

أظهرت النتائج التي تم الحصول عليها أن كل طريقة تقدر مثل هذه البيانات على أنها حالة شاذة أم لا وأن الطريقة المستندة إلى SVM من فئة واحدة أظهرت أدائها مقارنة بالآخرين.

الكلمات المفتاحية: التعلم غير الخاضع للإشراف ، شبكات الاستشعار اللاسلكية ، عزل الغابات ، SVMمن فئة واحدة ، K-Means , KNIME