



République Algérienne Démocratique et Populaire

Université Abou Bakr Belkaid– Tlemcen

Faculté des Sciences

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

**L'utilisation de DEEP clustering pour la surveillance non intrusive
des équipements électriques dans les maisons intelligentes**

Réalisé par :

- **DIABI Heythem**

Présenté le 22 Juin 2020 devant le jury composé de :

Président : - *Mr BENMAMMAR Badr*

Encadrant : - *Mr LEHSAINI Mohamed*

Examineur : - *Mr BENAMAR Abdelkrim.*

Co-encadrant : - *Mr KADDOUR Sidi Mohammed*

Année Universitaire : 2019-2020

Remerciements

A l'issue de ce travail, nous remercions, en premier lieu, le bon Dieu

De nous avoir donné la force et le courage de le mener à terme.

Nous tenons, également, à exprimer notre sincère reconnaissance et

Notre profonde gratitude à notre encadreur

Dont les conseils et orientations nous ont été

Précieusement utiles.

À tous les membres du jury pour l'honneur qu'ils nous font en

Acceptant de juger notre travail.

À tous nos enseignants et nos camarades de promotion

« 2019-2020 ».

Et à toute personne ayant contribué à l'élaboration de notre projet de

Près ou de loin.

Table des matières

Liste des figures	v
Liste des tableaux	vi
Acronymes	vii
Introduction générale.....	1
Chapitre I Les réseaux électriques intelligents (smart grids).....	3
I.1 Introduction	3
I.2 Les réseaux électriques intelligents.....	4
I.2.1 Définition.....	4
I.2.2 Gestion des smart grids.....	5
I.2.3 Principales caractéristiques des smart grids	5
I.2.4 La structure des smart grids.....	6
I.2.5 Les objectifs des réseaux électriques intelligents	7
I.2.6 Les bénéfices attendus des réseaux électriques intelligents	7
I.2.7 Les tâches intégrées des réseaux électriques intelligents	8
I.2.8 Standardisation dans les réseaux électriques intelligents	9
I.3 Les centrales de production de l'énergie électrique	10
I.3.1 Le réseau de transport.....	11
I.3.2 Le réseau de distribution.....	12
I.4 Conclusion.....	12
Chapitre II Systèmes de surveillance de l'énergie domestique.....	13
II.1 Introduction.....	13
II.2 Définition Maison intelligente	14
II.3 Systèmes de surveillance de domestique (HMS).....	14
II.3.1 Définition	14
II.3.2 HMS individuel	14

II.3.3 HMS partiel	14	
II.3.4 HMS et IoT (Internet des objets).....	15	
II.3.5 Architecture générale des systèmes de HMS	15	
II.4 Gestion Technique et Énergétique des Bâtiments (HEMS).....	16	
II.4.1 Architecture générale d'un système HEMS.....	17	
II.4.2 Quelques exemples d'application	19	
II.5 Types de mécanismes de prélèvement de la consommation d'énergie.....	20	
II.5.1 Surveillance d'énergie intrusive	20	
II.5.2 Surveillance d'énergie non intrusive	21	
II.5.3 Surveillance de charge hybride	23	
II.6 Conclusion	23	
Chapitre III	L'apprentissage automatique (Machine Learning).....	24
III.1 Introduction	24	
III.2 Machine Learning (ML)	24	
III.2.1 Préambule	24	
III.2.2 Le paradigme Machine Learning	25	
III.2.3 L'histoire du Machine Learning	25	
III.3 Concepts de l'apprentissage automatique.....	25	
III.3.1 Qu'est-ce que l'apprentissage artificiel	26	
III.3.2 Modélisation	26	
III.3.3 Domaines d'applications de l'apprentissage automatique.....	27	
III.4 Types d'apprentissage	27	
III.4.1 Apprentissage Supervisé	27	
III.4.2 Apprentissage par renforcement	27	
III.4.3 Apprentissage non supervisé.....	28	
III.4.4 Apprentissage semi-supervisé.....	28	
III.5 Les algorithmes utilisés dans l'apprentissage automatique.....	28	

III.6 Facteurs de pertinence de l'apprentissage automatique	29
III.7 Différentes étapes de l'apprentissage automatique	29
III.7.1 Ensemble de données	30
III.7.2 Entraînement ou apprentissage	32
III.7.3 Evaluation du modèle	32
III.7.4 Exploitation.....	33
III.8 Quelques méthodes de classification Automatique (Clustering).....	34
III.8.1 La méthode K-Means.....	34
III.8.2 Méthodes hiérarchiques	34
III.9 Introduction au Deep Learning.....	35
III.9.1 Définition de l'apprentissage en profondeur	36
III.9.2 Domaines d'application de deep learning.....	36
III.9.3 Différents types de modèles de deep learning	37
III.8 Conclusion	42
Chapitre IV Application : Classification automatique des évènements.....	43
IV.1 Introduction	43
IV.2 Environnement du développement.....	43
IV.2.1 La plateforme KNIME.....	43
IV.2.3 La bibliothèque TensorFlow	45
IV.2.4 La bibliothèque keras.....	46
IV.2.5 La bibliothèque Scikit-Learn	46
IV.2.6 La bibliothèque Numpy	46
IV.3 Collection des données	47
IV.3.1 BLUED Dataset	47
IV.3.2 Extraction des évènements.....	48
IV.4 Prétraitement des données	49
IV.4.1 Fast Fourier Transform	49

IV.4.2 Normalisation des valeurs.....	50
IV.5 Etapes D'apprentissage	51
IV.5.1 Autoencodeur pré-entraîné	51
IV.5.2 Modèle de classification automatique	52
IV.6 Evaluation.....	55
IV.7 Conclusion.....	56
Conclusion générale	57
Bibliographie.....	58

Liste des figures

Figure I-1: Illustration d'un réseau électrique intelligent [1]	5
Figure I-2: Les fonctions intégrées des réseaux électriques intelligents [5]	9
Figure I-3: Répartition de la production d'électricité dans le monde par type d'énergie primaire	10
Figure I-4: Répartition de la production d'électricité par type d'énergie primaire en France... ..	11
Figure I-5: Le réseau de transport maillé en France [5]	12
Figure II-1: Architecture générale d'un système HMS [7]	16
Figure II-2: Différence entre les systèmes de gestion de l'énergie dans les bâtiments [9].....	16
Figure II-3: Architecture générale d'un système HEMS [9]	17
Figure II-4: Consommation annuelle d'énergie dans un bâtiment commercial [10]	19
Figure II-5: Différentes catégories de charges basées sur les schémas énergétiques.....	22
Figure III-1: Schéma de modélisation d'une machine d'apprentissage	26
Figure III-2: Schéma d'un modèle supervisé.....	27
Figure III-3: Schéma d'un modèle non supervisé.....	28
Figure III-4: Schéma d'un modèle semi-supervisé.....	28
Figure III-5: Entraînement d'une machine d'apprentissage	32
Figure III-6: La relation entre l'intelligence artificielle, le ML et le deep Learning	36
Figure III-7: L'architecture des Auto-encodeurs	39
Figure III-8 : Diagramme du modèle DEC	40
Figure III-9 : La distribution auxiliaire P est dérivée de Q, mais plus concentrée autour des centroïdes.	41
Figure IV-1: Interface de la plateforme KNIME.....	44
Figure IV-0-2: Base de données NILM.....	47
Figure IV-3: Un exemple d'une fenêtre extraite d'un évènement Current.....	49
Figure IV-4: Le nœud Fast Fourier Transform (FFT).....	50
Figure IV-5: Une illustration d'un FFT	50
Figure IV-6: Le noeud Normalizer.....	50
Figure IV-7 Une illustration d'un autoencodeur.....	51
Figure IV-8: Le code source et le schéma de l'autoencodeur.....	52
Figure IV-9: Code de la couche personnalisée.....	53
Figure IV-10: Extrait du code de la distribution cible	55

Figure IV-11: Résultats de la classification automatique..... 56

Liste des tableaux

Tableau I-1: Niveaux de tension et répartition entre transport et distribution en France..... 6
Tableau IV-1: Caractéristiques du dataset BLUED 48

Acronymes

SGs	Smart Grids
TIC	Technologies de l'Information et de la Communication
IEC	Commission Electrotechnique Internationale
HTA/HTB	Haute Tension A/ Haute Tension B
HMS	Systèmes de surveillance de domestique
IoT	Internet des objets
HEMS	Gestion Technique et Énergétique des Bâtiments
RCSF	Réseaux de Capteurs Sans Fil
CVC	Chauffage, Ventilation et Climatisation
NILM	Non Intrusive Load Monitoring
ML	Machine Learning
CNN	Réseaux de neurones convolutionnels
DBN	Deep Belief Network
DEC	Clustering intégré en profondeur
DNN	Deep Neural Networks
KNIME	Konstanz Information Miner
SVMs	Machines à Vecteurs de Support
KL	Divergence Kullback – Leibler

Résumé

Les solutions de surveillance de l'énergie dans les maisons intelligentes proposées ces dernières années convergent vers des méthodes d'apprentissage non supervisées qui peuvent être adaptables et scalables.

Dans le cadre de ce travail, nous avons proposé une solution qui permet de détecter les états des équipements en traitant seulement la charge électrique agrégée. Pour cela, nous avons utilisé la méthode d'apprentissage approfondi (DEC : Deep Clustering). Cette méthode de classification automatique se base sur un apprentissage approfondi non supervisé. Nous avons pris comme environnement de validation une maison surveillée dans laquelle il y a un ensemble d'équipements (une machine à laver, une lave-vaisselle, un chauffage et un climatiseur, etc). Par ailleurs, le but de la solution proposée est de créer des compteurs intelligents qui permettent de connaître l'équipement qui est actif à un instant donné en fonction de la consommation de l'électricité. Cette solution se base sur un nouveau mécanisme de clustering des images qui permet la réduction de la dimension du problème traité en utilisant les "AutoEncoders" et la plateforme KNIME.

Mots Clés : Deep Clustering, KNIME, AutoEncoders, Apprentissage non supervisé, Réseaux de capteurs.

Abstract

Smart home energy monitoring solutions provided in recent years converge on unsupervised learning methods that can be adaptable and scalable.

In this work, we proposed a solution which makes it possible to detect the states of the equipment by treating only the aggregated electrical charge. For this, we used the deep learning method (DEC: Deep Clustering). This automatic classification method is based on in-depth unsupervised learning. We used as a validation environment a monitored house in which there is a set of equipments (a washing machine, a dishwasher, a heater and an air conditioner, etc.). In addition, the goal of the proposed solution is to create smart meters which allow to know the equipment which is active at a given time according to the consumption of electricity. This solution is based on a new image clustering mechanism which allows the reduction of the dimension of the problem treated by using "AutoEncoders" and the KNIME platform.

Keywords: Deep Clustering, KNIME, AutoEncoders, Unsupervised learning, Sensor networks.

ملخص

تتقارب حلول مراقبة الطاقة في المنازل الذكية المقترحة في السنوات الأخيرة على طرق التعلم غير الخاضعة للإشراف والتي يمكن أن تكون قابلة للتكيف وقابلة للتطوير.

في إطار هذا العمل ، اقترحنا حلاً يجعل من الممكن اكتشاف حالات المعدات من خلال معالجة الشحنة الكهربائية المجمعة فقط. لذلك ، استخدمنا طريقة التعلم العميق (DEC: Deep Clustering). تعتمد طريقة التصنيف التلقائي المستعملة على التعلم المتعمق غير الخاضع للرقابة. استخدمنا كبنية للتحقق من الصحة منزلاً مراقباً توجد فيه مجموعة من المعدات (غسالة ، غسالة صحون ، سخان ومكيف هواء ، إلخ). بالإضافة إلى ذلك ، فإن الهدف من الحل المقترح هو إنشاء عدادات ذكية تسمح بمعرفة المعدات النشطة في وقت معين وفقاً لاستهلاك الكهرباء. يعتمد هذا الحل على آلية جديدة لتجميع الصور تسمح بتقليل أبعاد المشكلة التي يتم معالجتها باستخدام "AutoEncoders" ومنصة KNIME.

الكلمات المفتاحية: التجمعات العميقة، KNIME، الترميز التلقائي، التعلم بدون إشراف، شبكات أجهزة الاستشعار.

Introduction Générale

Introduction générale

La consommation d'énergie dans le monde connaît une augmentation sans cesse. Elle est d'environ 2% par an, alors que l'on assiste à une réduction des ressources énergétiques de type fossiles. D'où, il apparait un risque majeur concernant l'avenir de notre planète du fait du changement climatique. L'énergie va donc devenir une ressource précieuse (rare) et plus coûteuse. C'est pour cette raison la maîtrise des consommations énergétiques ainsi que le recours aux énergies renouvelables, prennent plus d'importance ces dernières années. Il faut à ce sujet noter que dans le programme de la demande énergétique, le gain d'un MWh obtenu en améliorant le rendement d'utilisation et les techniques d'exploitation nécessite seulement 13.5 € d'investissement tandis que la production d'un MWh coûte plus de 32.5€.

Par exemple, en 2008, le domaine du bâtiment (résidentiel ou autre) est le domaine qui a consommé plus d'énergie primaire soit 69.4 MTep (43.4%) de la consommation totale d'énergie primaire (160 MTep) en France. En 2009, la consommation d'électricité du secteur résidentiel et tertiaire est arrivée à 293.8 TWh, soit 64.4% de l'électricité consommée (456.3 TWh).

Notre objectif est de réduire la consommation d'électricité dans les bâtiments. Pour cela plusieurs voies peuvent être explorées, telles que la mise au point d'enveloppes thermiques hyper-isolées pour les bâtiments ou le développement des systèmes de production d'énergie à haut rendement. De nombreux efforts ont été réalisés afin de trouver la meilleure approche pour gérer la consommation d'énergie dans les bâtiments tout en protégeant l'environnement. Cet objectif est explicitement affiché dans une directive de performance énergétique, qui précise l'importance de l'économie des ressources énergétiques et de la réduction des émissions de CO₂. Dans cette optique, il devient important de mener une stratégie pour maîtriser la demande d'électricité. Cette stratégie vise à gérer d'une manière optimale la consommation de l'électricité dans les bâtiments. Ainsi, la régulation et la gestion intelligente de la consommation de l'électricité sont une des préoccupations prépondérantes pour les gestionnaires, les fournisseurs et les consommateurs d'énergie.

Dans le cadre de ce projet de fin d'études, nous avons proposé une solution qui consiste à détecter les états des équipements en traitant seulement la charge électrique agrégée. Pour ce faire, nous avons utilisé la méthode d'apprentissage approfondi (DEC). Cette méthode de classification automatique se base sur un apprentissage approfondi non supervisé. Nous avons pris comme

environnement de test une maison surveillée dans laquelle il y a un ensemble d'équipements tels qu'une machine à laver, un lave-vaisselle, un chauffage et un climatiseur, etc. Par ailleurs, le but de la solution proposée est de créer des compteurs intelligents (soft sensors) qui permettent de connaître l'équipement qui est actif à un instant donné en fonction de la consommation de l'électricité.

Le manuscrit est organisé en quatre chapitres :

Dans le premier chapitre, nous présentons les "Smart Grids", leurs caractéristiques et leurs valeurs ajoutées en faveur du réseau électrique actuel.

Dans le deuxième chapitre, nous décrivons les systèmes de surveillance de l'énergie électrique et la gestion technique et énergétique dans les bâtiments.

Dans le troisième chapitre, nous présentons l'apprentissage automatique (Machine Learning) et leurs concepts fondamentaux. Puis, nous étalons également les principales approches de classification en mettant l'accent sur leurs avantages et leurs limitations.

Le quatrième chapitre est consacré à la solution que nous avons proposée dans le cadre de ce projet de fin d'études. Cette solution consiste en application permettant de détecter les événements (connaître les équipements actifs au sein d'un habitat). Pour ce faire, nous avons opté pour une méthode de classification (DEC).

Enfin, nous terminons ce manuscrit par une conclusion générale dans laquelle nous résumons la démarche suivie pour mener à une bonne fin 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 a été développé de telle sorte qu'il fonctionne selon un schéma hiérarchique : production, transport, et distribution. Il est baptisé sur des dispositifs lui permettant à maintenir sa fiabilité et son efficacité. Cependant, aujourd'hui le réseau électrique dans cette structure fait face à des défis imposés par l'intégration des énergies renouvelables, l'évolution technologique dans ce domaine, et les exigences des utilisateurs finaux. Ainsi, le réseau électrique du futur, appelé "Smart Grid (SG)", sera basé sur les techniques issues de la technologie de l'information et de la communication (TIC) et les dispositifs de mesure en temps réel. Ceci permettra d'améliorer la résilience et les prévisions des consommations des partenaires. Par ailleurs, le nouveau paradigme "Smart Grid" se base sur le dégroupage du réseau, sa restructuration et l'optimisation de ses dispositifs.

La diversité des installations de production des énergies renouvelable a donné naissance à de nouveaux défis pour les producteurs de l'électricité. Ces défis concernent essentiellement l'intégration de ces sources diversifiées dans un réseau électrique existant. C'est dans cette optique que le paradigme "Smart Grid" s'est imposé puisqu'il permet le mariage entre le réseau électrique actuel et les méthodes intelligentes pour sa gestion (production et exploitation). Ce mariage a donné naissance à nouveau type de réseaux électriques appelé "les réseaux électriques intelligents".

Dans ce chapitre, on présente le paradigme "Smart Grid", ses caractéristiques et sa valeur ajoutée en faveur du réseau électrique actuel.

I.2 Les réseaux électriques intelligents

I.2.1 Définition

Le Smart Grid est un réseau électrique intelligent qui utilise les technologies de l'information et de la communication (TIC) pour optimiser la production, la distribution et la consommation d'électricité et garantir l'efficacité et la fiabilité du réseau électrique mis en place.

Les smart grids sont des nouvelles architectures du réseau électrique qui permettent l'intégration des sources d'énergies renouvelables de différentes natures. Ils contiennent des dispositifs du réseau électrique traditionnel tels que les postes de transformation, les systèmes de protection, ainsi que les systèmes d'intégration des énergies renouvelables et les technologies de communication (WiFi, 4G, et éventuellement la 5G dans le futur...). Ces deux derniers aspects qui ont permis de basculer d'un réseau électrique traditionnel à un réseau électrique intelligent.

Les réseaux électriques intelligents permettent d'intégrer d'une manière efficace les opérations des producteurs et des consommateurs de l'électricité dans le but est d'assurer un approvisionnement durable et à moindre coût de l'électricité. En outre, les nouvelles technologies permettront de faire des économies en utilisant la production décentralisée et impliquer le consommateur dans la gestion de la consommation de l'électricité. Par exemple, instaurer trois états de consommation (rouge, orange et vert). L'état rouge correspond à un pic de consommation et dans ce cas le consommateur est sensé par exemple d'éteindre les équipements ou les lampes qui ne sont pas très indispensables pour lui ou réduire leur consommation en électricité par exemple augmenter la température d'un climatiseur de quelques degrés. L'état orange correspond à une consommation moyenne qui ne pourra pas perturber le réseau électrique alors que l'état vert correspond à une consommation faible. De cette manière l'approvisionnement des consommateurs en électricité sera plus fiable et le coût de l'électricité pourrait être réduit. La Figure I-1 montre le schéma de principe d'un smart grid [1].

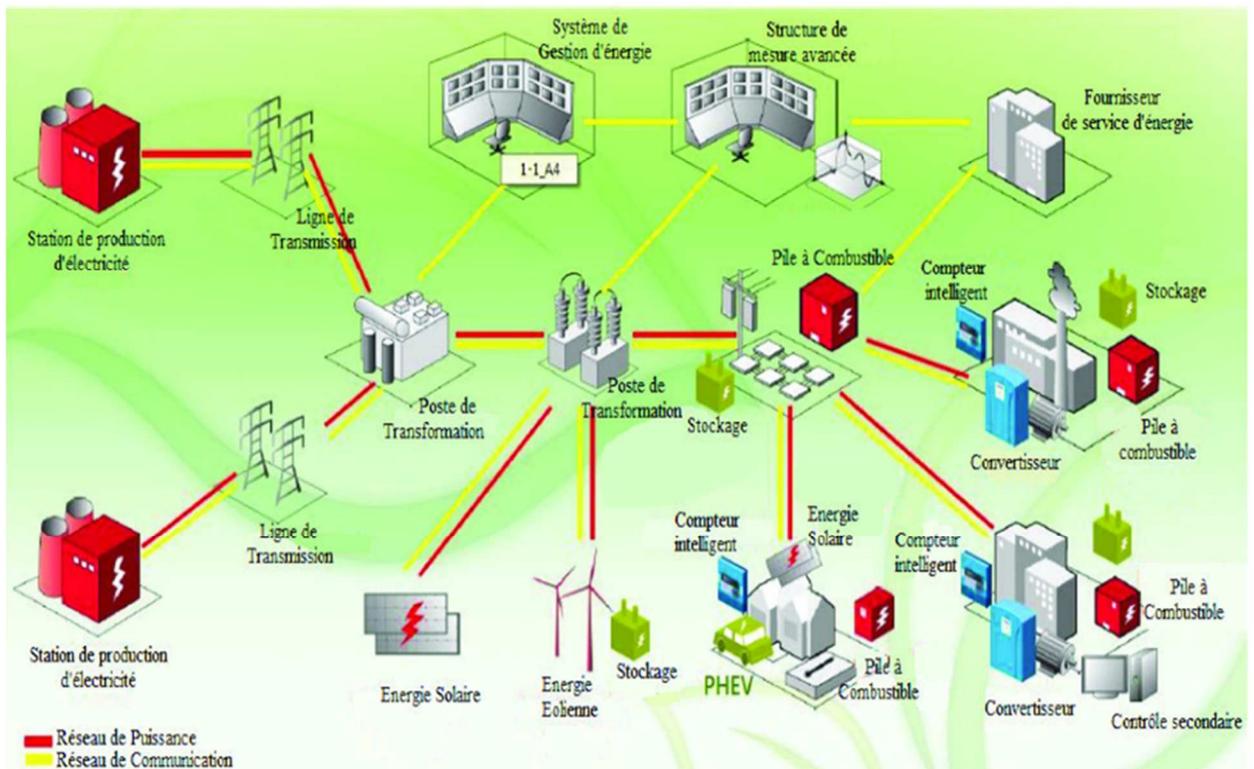


Figure I-1: Illustration d'un réseau électrique intelligent [1]

I.2.2 Gestion des smart grids

Dans le réseau électrique traditionnel, la gestion est faite d'une manière centralisée et souvent l'opérateur n'a pas d'information sur la distribution de l'énergie électrique dans le réseau. Dans certaines situations, l'opérateur ne sait pas catégoriser l'état de la consommation de l'énergie électrique et par conséquent s'il s'agit d'un pic de consommation on pourra se retrouver avec des clients qui seront prévus en électricité et de même ne pourra pas connaître si un client est prévu de l'électricité ou non. Cependant, avec les smart grids cette information sera disponible par le biais du compteur intelligent de l'abonné en utilisant les technologies de l'information et de la communication. Ainsi par l'intégration des TIC, d'autres visions sur l'état du réseau électrique s'ouvrent en particulier le contrôle du réseau électrique en temps réel et également le client pourra être impliqué d'une certaine manière à la stabilité du réseau électrique en fonction de la consommation globale dans le réseau électrique. De ce fait, à travers ce paradigme de communication, on aboutit à une structure bidirectionnelle du réseau électrique [2].

I.2.3 Principales caractéristiques des smart grids

Les smart grids se caractérisent principalement par :

- Auto-recouvrement : Les SGs ont la capacité de recouvrement s'il y a l'occurrence d'une perturbation ou une panne dans le réseau.
- Permettre l'implication du consommateur dans la gestion de l'énergie par le biais des compteurs intelligents.
- Résilience contre d'éventuelles attaques physiques ou cyberattaques.
- Permettre l'augmentation de la capacité de réception de nouvelles sources de production de l'électricité.
- Permettre la production de l'énergie électrique selon le besoin (répondre à la demande).

Ces deux dernières caractéristiques ont un grand intérêt puisqu'il est difficile de stocker l'énergie électrique. Dans ce cas de figure, il y a un compromis entre produire plus d'énergie que le nécessaire et dans ce cas il y aura une perte de l'énergie produite ou produire moins et dans ce cas on se trouve devant une situation où le client ne sera pas satisfait par manque d'approvisionnement en électricité comme c'est arrivé en 2018 (une forte demande durant l'été 2018 lors de la canicule) [3].

I.2.4 La structure des smart grids

Les réseaux électriques permettent d'acheminer l'énergie électrique à partir des centres de production jusqu'aux consommateurs tout en assurant une livraison fiable. Dans ce type de réseaux, les pertes sur le réseau sont essentiellement dues au phénomène de l'effet Joule au niveau des conducteurs. Par ailleurs, réduire les pertes causées par ce phénomène au niveau des conducteurs est une opération très onéreuse, puisque cela revient à utiliser des matériaux présentant une plus faible résistivité. Ainsi, la solution adoptée pour surpasser cette anomalie est d'élever le niveau de la tension pour que le courant circulant dans les conducteurs soit plus faible. Le tableau I-1 présente les niveaux de tension sur le réseau électrique en France et la répartition entre le réseau de transport et le réseau de distribution [4].

Tableau I-1: Niveaux de tension et répartition entre transport et distribution en France

Niveaux de tension	Appellation d'après la norme	Gestionnaire	Kilomètres en France
400 KV	HTB	Transport	100 000 KM
225 KV	HTB	Transport	
90 KV	HTB	Transport	
63 KV	HTB	Transport	
20 KV	HTA	Distribution	1.3 million KM
400 KV	BT	Distribution	

I.2.5 Les objectifs des réseaux électriques intelligents

Les Etats-Unis est le premier pays qui a lancé le projet des *smart grids* en 2009 et depuis plusieurs pays ont investis sur ce projet. Ainsi, à travers les réseaux électriques intelligents, les différents opérateurs ont visé plusieurs objectifs sur plusieurs plans : économique, environnemental, et social. Par exemple, les industriels du Gimélec¹ et leurs partenaires avait une vision technologique, marketing et économique pour pour générer des bénéfices économiques, environnementaux et sociétaux.

En outre, la protection de l'environnement et du climat combinée à la rareté des énergies fossiles dans le futur ainsi que la sécurité énergétique, ont motivé les opérateurs dans le domaine et les gouvernements de faire recours aux énergies renouvelables (le vent, le soleil, la terre et la mer). Par ailleurs, compte tenu de la volatilité de la production des énergies renouvelables causée par plusieurs tels que le cycle diurne/nocturne et les conditions météorologiques, il est indispensable d'instaurer un système permettant l'instauration de la disponibilité et la flexibilité de la production d'électricité. La fourniture de l'énergie électrique (énergies conventionnelles et renouvelables) doit répondre aux besoins de la demande des consommateurs avec une adéquation entre l'offre et de la demande [5].

I.2.6 Les bénéfices attendus des réseaux électriques intelligents

Les bénéfices attendus des réseaux électriques intelligents engendrent plusieurs volets [5].

a) Pour les gouvernements

- Mise en place des conditions favorables à l'efficacité énergétique.
- Augmentation de l'indépendance énergétique par l'intégration des énergies renouvelables.
- Augmentation de la sécurité des réseaux électriques.
- Implication des nouvelles technologies pour la production de l'énergie électrique et soutenir la recherche nationale.
- Contribution à la préservation des matières premières.

b) Pour les consommateurs

- Atteindre des solutions d'efficacité énergétique compétitives existantes.
- Promouvoir un bon comportement avec une tarification plus flexible selon l'usage.

¹ Gimélec : Groupement des entreprises de la filière électronique française

- Modification des comportements tout en respectant la vie privée des consommateurs selon une législation spécifique sur la protection des données.
- Meilleur contrôle de la production d'énergie renouvelable et de nouveaux usages.

c) Pour les distributeurs d'électricité

- Optimisation de la distribution d'électricité à travers des réseaux électriques mis en place.
- L'intégration de l'utilisation des voitures électriques et des nouveaux usages électriques.
- Assurer l'équilibre du réseau pour les entreprises chargées du transport de l'énergie.
- Gestion complète du système de production, de transport et de distribution
- Gestion par prévision des capacités disponibles de production d'énergies en fonction de la demande.

d) Pour les industriels en électricité

- Durabilité des emplois à forte valeur ajoutée dans le domaine de la production de l'électricité.
- Soutenir l'exportation du savoir-faire dans le domaine de la production de l'électricité à travers une interconnexion accrue des réseaux.
- Promouvoir les critères de normalisation.

e) Pour l'université et les centres de recherche

- Mobilisation et soutien des chercheurs dans le domaine des Smart Grids sur tous les plans (production, transport et distribution).
- Promouvoir les projets d'innovation dans le domaine de l'énergie électrique.

I.2.7 Les tâches intégrées des réseaux électriques intelligents

Les principales fonctions intégrées dans les réseaux électriques intelligents sont les suivantes (Voir Figure I-2) :

- Production de l'énergie électrique d'une manière conventionnelle ou renouvelable.
- Qualité et optimisation du transport de l'énergie électrique.
- Production, automatisation, sécurisation et contrôle des réseaux électriques.
- Pilotage global des réseaux électriques intelligents sur tous les plans de la production à la distribution.
- Stockage de l'électricité dans plusieurs régions.
- Gestion informatique des données.

- Système de prélèvement automatique.
- Implication du consommateur dans la gestion des réseaux électriques intelligents.
- Intégration des véhicules électriques et d'autres usages.

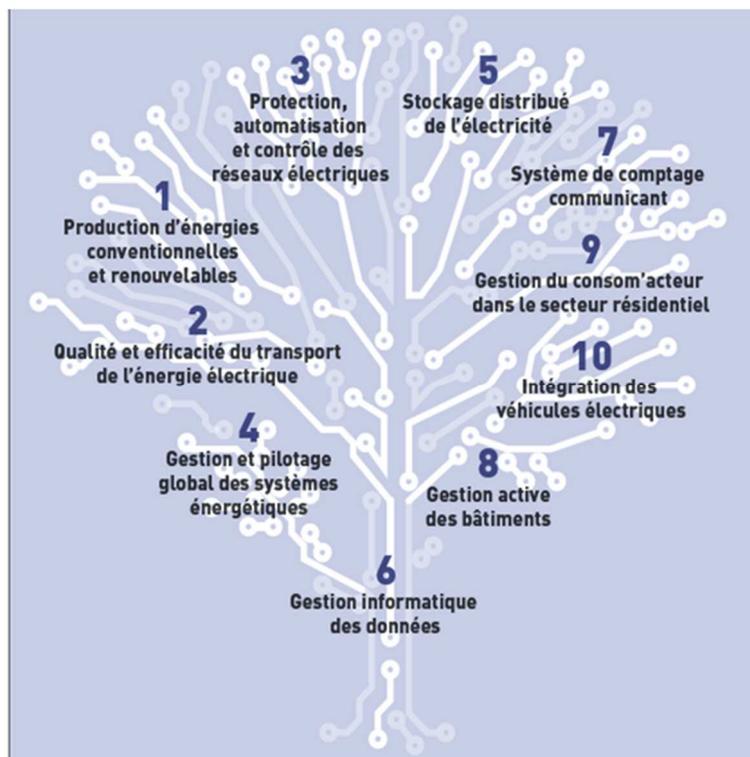


Figure I-2: Les fonctions intégrées des réseaux électriques intelligents [5]

I.2.8 Standardisation dans les réseaux électriques intelligents

La standardisation dans les réseaux électriques intelligents est primordiale pour développer cette nouvelle technologie. Ainsi les pouvoirs publics et les opérateurs dans le domaine de l'énergie électrique doivent travailler ensemble pour standardiser (normaliser) les fonctions dans ce secteur. Ainsi, cette standardisation doit concerner l'atteinte de l'optimum économique du marché, la mise en réseau du système électrique et les législations juridiques des investissements.

Par exemple en France, les actions prioritaires en matière de standardisation ont été engagées par la Commission Electrotechnique Internationale (IEC). Cette commission avait toutes les prérogatives pour élaborer des normes ce qui permet d'éviter les redondances. Parmi les actions prioritaires en matière de standardisation (normalisation) associées à la commission IEC figurent notamment :

- Accompagner les processus déjà mis en place,
- Harmoniser les mesures d'énergie ainsi que leur contrôle selon des règles bien précises.

- Accroître la collaboration avec les nouvelles technologies de l'information et de la communication,
- Standardiser des règles de la protection de la vie privée et des données personnelles [6].

I.3 Les centrales de production de l'énergie électrique

La production d'électricité est faite d'une manière centralisée. Elle est généralement produite par de grandes centrales se trouvant éloignées des zones urbaines et connectées au réseau de transport via des transformateurs élévateurs. Ces grands alternateurs sont généralement des machines entraînées par des turbines. Les principales sources d'énergie utilisées par ces turbines sont : les énergies fossiles, nucléaires ou hydrauliques. La capacité de production de ces centrales peut aller de quelques centaines de mégawatts pour les petites installations à gaz ou à charbon, à 1650 MW pour les centrales nucléaires. Les centrales de production peuvent être distinguées par les sources d'énergie primaires utilisées pour faire fonctionner leurs alternateurs. La Figure I-3 présente la répartition des centrales en activité dans le monde en 2012.

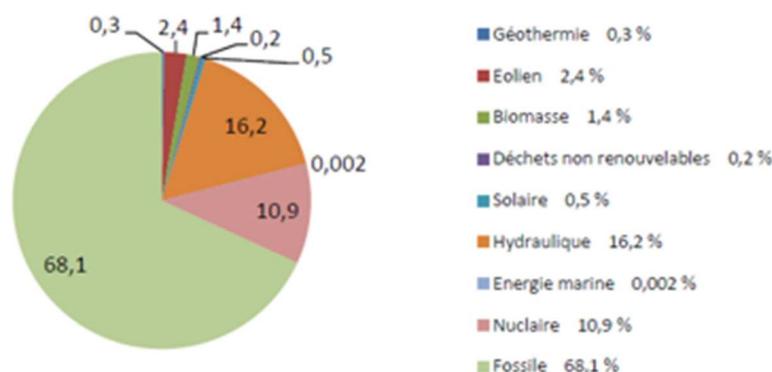


Figure I-3: Répartition de la production d'électricité dans le monde par type d'énergie primaire

Par ailleurs, cette répartition de la production d'électricité se diffère d'un pays à un autre selon ses ressources naturelles dont il dispose et ses compétences dans ce domaine. Par exemple, en France cette répartition est donnée comme c'est illustré dans la Figure I-4.

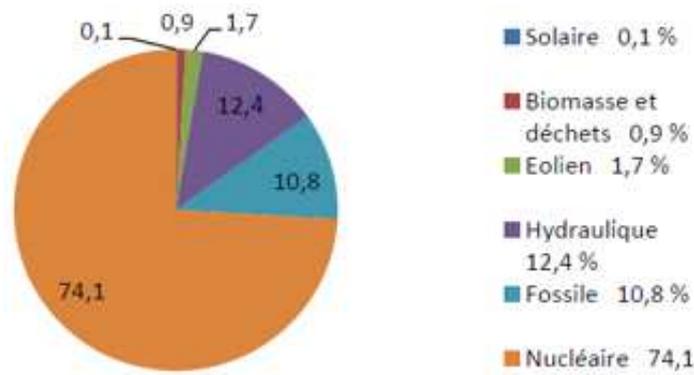


Figure I-4: Répartition de la production d'électricité par type d'énergie primaire en France

La stabilité du réseau électrique est principalement garantie par la bonne gestion de ces centrales de production. Ces centrales de production sont gérées d'une manière synchrone selon trois niveaux de réglages : primaire, secondaire et tertiaire. Ces niveaux de réglage permettent d'assurer un équilibre entre la production et la consommation tout en répartissant la réserve de puissance entre les centrales de production impliquées dans ce processus. Cette réserve de puissance permet de répondre à tout instant à une augmentation de la consommation.

I.3.1 Le réseau de transport

Le réseau de transport permet la liaison entre les centrales de production et le réseau de distribution. Il achemine l'énergie électrique sur des grandes distances à des niveaux de tension élevés afin de minimiser les pertes par effet Joule. Par exemple, en France, le transport est réalisé à l'aide de quatre niveaux de tension comme c'est indiqué dans tableau 1 pour un maximum de 400 kV. Dans d'autres pays plus étendus, le transport est réalisé sur des distances de plusieurs milliers de kilomètres et les niveaux de tension peuvent atteindre jusqu'à 800 kV, voire 1 MV. Par ailleurs, dans le but de garantir la robustesse du réseau de transport, ce dernier est présenté selon une architecture maillée comme c'est illustré dans la Figure I-5 où les flux de puissance sont répartis sur toutes les lignes du réseau. En outre, ces flux sont différents entre les zones de production et les zones de consommation et par conséquent leur gestion est une tâche complexe.

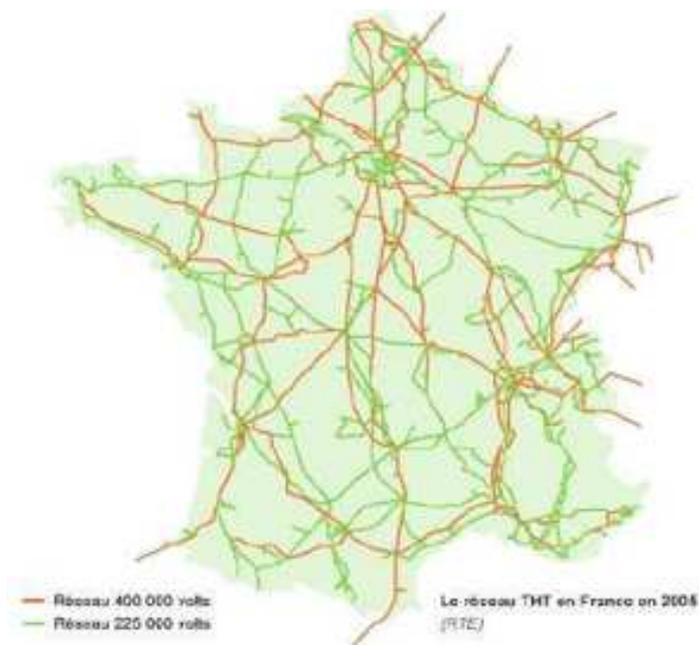


Figure I-5: Le réseau de transport maillé en France [5]

I.3.2 Le réseau de distribution

Le réseau de distribution permet d'alimenter les zones de consommation. Il commence à partir du poste de transformation HTB/HTA², qui représente la frontière entre les deux réseaux (le réseau de transport et le réseau de distribution). A partir de ce poste, les conducteurs prennent des directions différentes pour couvrir les zones de consommation. Cette distribution est faite à un niveau de tension inférieur à 50 kV, et se divise en deux réseaux de type HTA dont le niveau de tension est généralement de 20 kV, et le réseau BT (basse tension) à 400 V.

I.4 Conclusion

Dans ce premier chapitre, on a présenté le contexte de ce travail et plus particulièrement les réseaux électriques intelligents (les Smart Grids). On a mis l'accent sur les spécificités des réseaux électriques intelligents comparativement aux réseaux électriques conventionnels. Puis, on a présenté les caractéristiques de ce nouveau type de réseaux, leurs fonctions principales pour améliorer le domaine de l'énergie électrique sur tous les plans : production, transport et distribution.

Dans le chapitre suivant, on va s'intéresser principalement aux systèmes de surveillance de l'énergie domestique c'est-à-dire on s'intéresse à la surveillance de la consommation de l'électricité au sein des habitats.

² HTA/HTB : Haute Tension A/ Haute Tension B

Chapitre II

Systemes de surveillance de

L'énergie domestique

Chapitre II

Systemes de surveillance de l'énergie domestique

II.1 Introduction

Les anciens bâtiments ont été construits par des abris primitifs (pierre, bâtons, peau d'animal et d'autres matériaux naturels) dans le but est de fournir un espace confortable aux occupants. Par ailleurs, les bâtiments d'aujourd'hui sont formés de structures complexes, de systèmes et de technologies. Ainsi, au cours du temps, chacun élément à l'intérieur d'un bâtiment a été développé pour permettre aux occupants de choisir d'une manière autonome les systèmes d'éclairage, de chauffage, de climatisation, et de sécurité au sein de l'habitat. En outre, Les occupants des habitats d'aujourd'hui sont sensés de prendre conscience de l'impact des nouvelles constructions sur l'environnement. Pour cela, dans les bâtiments du futur, il ne suffit pas de se contenter seulement des systèmes qui procurent le confort et la sécurité, mais d'intégrer des systèmes permettant de minimiser les coûts énergétiques sans avoir une influence sur l'environnement. Ces bâtiments sont dits bâtiments intelligents. Ils permettent de fournir des services utiles et intelligents tels que l'éclairage, le confort thermique, la qualité de l'air, et la sécurité, etc... Pour atteindre cet objectif, il faut introduire l'intelligence et la technologie de l'information et de la communication à plusieurs niveaux pour relier une variété de sous-systèmes, qui fonctionnent généralement d'une manière autonome entre eux, de sorte que ces systèmes puissent partager des informations pour maximiser la performance globale au sein du bâtiment. En outre, ce nouveau type de bâtiments a la capacité de fonctionner même au-delà de leurs limites physiques. Ils sont connectés entre eux et Internet ainsi qu'ils permettent de répondre au réseau électrique d'une manière intelligente. Ils interagissent également avec les occupants du bâtiment pour leur fournir des informations utiles.

Dans ce chapitre, on présente les systèmes de surveillance de l'énergie électrique et la gestion technique et énergétique dans les bâtiments.

II.2 Définition Maison intelligente

D'après F-X.Jeuland [7] : "Une maison intelligente est une maison qui dispose d'accommodations susceptibles de faciliter la vie de ses habitants au quotidien, de permettre des économies d'énergie et d'apporter un certain niveau de confort et de sécurité. Elle est expansive aux évolutions futures par le biais de ses infrastructures et par son ouverture au monde numérique".

II.3 Systèmes de surveillance de domestique (HMS)

II.3.1 Définition

Un système HMS un système qui contrôle automatiquement des dispositifs dotés de microcontrôleur ayant la capacité de communiquer entre eux via des protocoles standards, un système de contrôle-commande centralisé et une interface homme-machine permettant le suivi de la consommation d'énergie et certaines fonctions de perfectionnement pour la gestion de l'éclairage, la climatisation, le chauffage au sein d'un habitat [7].

II.3.2 HMS individuel

Un système de HMS individuel consiste à automatiser les tâches individuelles de chaque équipement au sein de l'habitat comme le contrôle de l'éclairage ou la climatisation sans tenir compte des événements externes. Dans ce système restreint, les occupants doivent recueillir des données à partir de divers endroits [8].

Un système HMS individuel est caractérisé par :

- Interconnectivité réduite.
- Le nombre des interventions manuelles de l'occupant au sein de l'habitat est réduit.
- La prise de décision n'est pas totale.
- Nécessité d'une infrastructure.
- Initiatives individuelles de développement durable.

II.3.3 HMS partiel

Un système de HMS partiel consiste à harmoniser quelques activités des équipements en visant en particulier d'économiser l'énergie à travers les systèmes de gestion de l'énergie. Comparativement au système de HMS individuel, ces systèmes sont plus impliqués, dans lesquels l'intervention humaine est très réduite, et permettent une prise de décisions.

Un système de HMS partiel est caractérisé par :

- Une interconnectivité plus importante à travers des solutions logicielles intégrées.
- Intervention humaine est très réduite dans la gestion des activités des équipements.
- Prise de décision très rapide grâce à sa vue globale du système tout entier.
- Amélioration des initiatives de durabilité comparativement à celles dans un système de HMS individuel [8].

II.3.4 HMS et IoT (Internet des objets)

L'intégration de l'aspect IoT dans les systèmes HMS a permis la création des données afin d'avoir une analyse plus approfondie et une prise de décision précise en temps réel sur le système et ses composants. Cette intégration est caractérisée par [8]:

- Communication plus étendue entre les périphériques, stockage intégré et analyse détaillée et précise de diverses informations sur les plateformes communes. Il permet aussi l'intégration d'un environnement cloud.
- Les interventions manuelles sont très réduites.
- La prise de décision est intelligente avec des décisions précises et automatisées.
- Amélioration de la base de connaissances dans le système.
- Tirer profit d'une seule infrastructure pour exploiter toutes les solutions HMS mises en place dans l'environnement surveillé.

II.3.5 Architecture générale des systèmes de HMS

Les réseaux de capteurs ont permis aux systèmes HMS de définir leurs principales composantes. Une architecture générale pour un tel système HMS est présentée par la Figure II-1. Cette architecture peut être devisée conceptuellement en trois éléments :

- Couche de capteur : Le bâtiment est surveillé par un ensemble de capteurs qui collectent des données sur un phénomène bien précis. Ces données sont exploitées par la couche de calcul pour être traitées et analysées.
- Couche de calcul : Fournit des informations traitées par le biais des calculs algorithmiques et des analyses statistiques, à la couche d'application afin qu'elle prenne la bonne décision.
- Couche application : Cette couche a deux tâches : le contrôle des équipements et la fourniture des informations aux utilisateurs concernant un objectif bien précis [7].

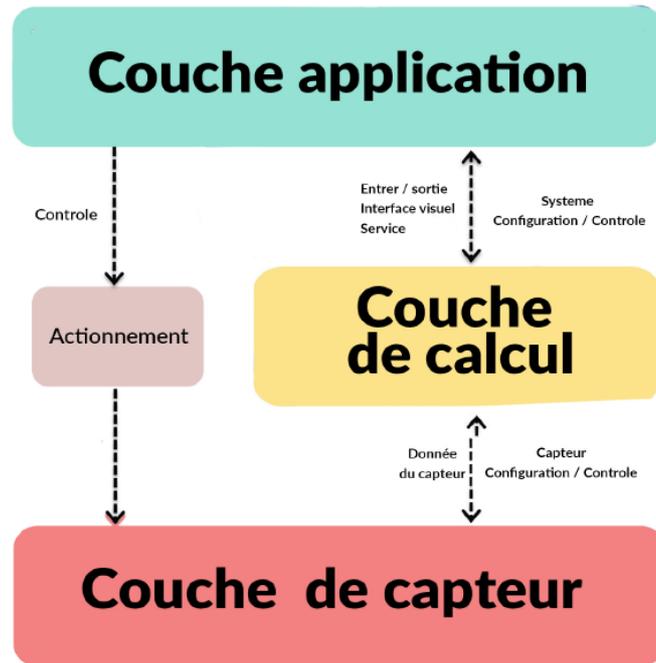


Figure II-1: Architecture générale d'un système HMS [7]

II.4 Gestion Technique et Énergétique des Bâtiments (HEMS)

Le système HEMS est un système de gestion des bâtiments qui s'occupe de la surveillance de l'environnement et des événements en impliquant les réseaux de capteurs sans fil (RCSF) afin de collecter des informations détaillées sur la consommation d'énergie dans les bâtiments. Par ailleurs, quand il s'agit d'un système HEMS amélioré avec une rétroaction activée, ce dernier contrôle l'énergie en temps réel afin de minimiser la consommation. Les systèmes HEMS représentent la pierre angulaire dans les bâtiments dits intelligents [9].

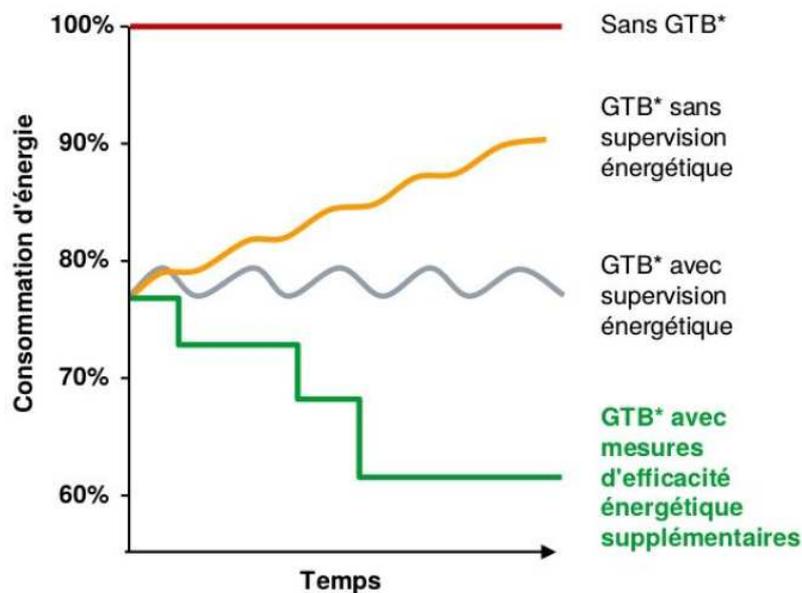


Figure II-2: Différence entre les systèmes de gestion de l'énergie dans les bâtiments [9]

II.4.1 Architecture générale d'un système HEMS

Ce type de systèmes est baptisé généralement sur une architecture à trois niveaux : un réseau de capteurs qui fournit la consommation d'énergie, le comportement des occupants de l'habitat et les conditions environnementales au sein de l'habitat ; un serveur central qui contrôle les appareils électriques dans un bâtiment. La première couche de cette architecture fournit des informations au serveur de contrôle. Ce dernier réalise le calcul sur les informations collectées par les capteurs ainsi que des informations reçus à partir d'autres sources, telles que les conditions météorologiques/prévisions en utilisant Internet. Ces données sont ensuite traitées par des algorithmes intelligents pour contrôler l'énergie en éteignant les équipements qui sont en état actif tels que les appareils électriques et les lampes [9] (Figure II-3).

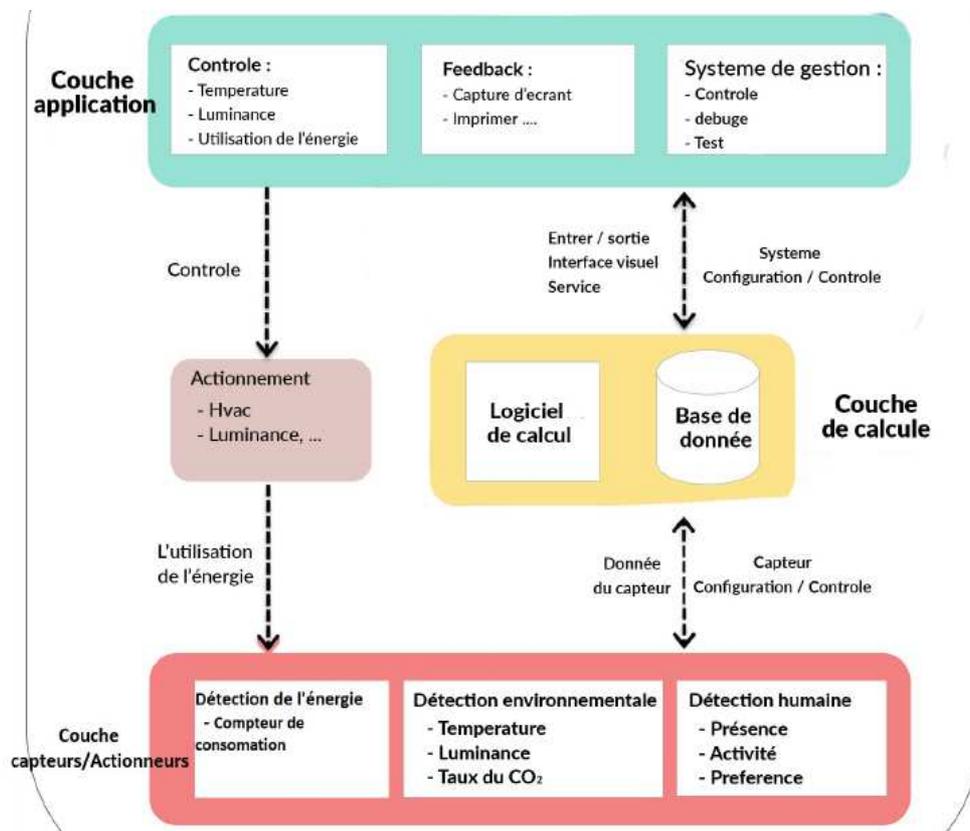


Figure II-3: Architecture générale d'un système HEMS [9]

a) Couche "capteurs/actionneurs"

La couche "capteurs/actionneurs" est une couche de niveau physique contenant un ensemble de capteurs déployés à l'intérieur d'un bâtiment qui mesurent périodiquement des phénomènes bien précis et envoient les données collectées vers la couche de calcul pour les analyser et les stocker. Cette couche peut être conçue avec trois sous-couches :

- Détection de l'énergie : cette sous-couche consiste à déployer des compteurs intelligents dans lesquels il y a des capteurs qui contrôlent la consommation de l'énergie dans un bâtiment.
- Détection environnementale : au niveau de cette sous-couche, un ensemble de capteurs est déployé pour recueillir des informations sur les facteurs environnementaux prédominants tels que la luminosité, la température, l'humidité, le niveau CO₂, etc.
- Détection humaine : au niveau de cette sous-couche des capteurs sont déployés pour détecter la présence des occupants, et élaborer des modèles de comportement pour un environnement donné.

b) Couche de calcul

La couche de calcul permet d'analyser toutes les informations fournies par la couche "capteurs/actionneurs". Elle comprend deux composantes :

- Logiciel de calcul : permet d'analyser les données reçues par la couche "capteurs/actionneurs" et de proposer une solution en se basant sur des raisonnements logiques, l'exploration des données et les besoins de la couche application.
- Base de données : stocke toutes les données en temps réel afin de faire une analyse plus tard tels que les comparaisons de consommation de l'énergie, l'historique des calculs et l'assemblage des modèles.

c) Couche application

La couche d'application permet de réaliser deux fonctions :

- Le retour d'information : cette fonction a pour rôle est de fournir aux occupants d'un habitat ou à une tierce partie des informations concernant la consommation d'énergie dans un bâtiment. Ces informations sur la consommation d'énergie peuvent être présentées en temps réel ou en consultant l'historique et ces informations peuvent être visualisées via des périphériques mobiles ou des sites Web.
- Contrôle d'énergie : dispositifs sont mis en place pour piloter des systèmes de gestion d'énergie dans le bâtiment selon les politiques et les préférences prédéfinies par les clients (consommateurs). Ces préférences peuvent être définies en utilisant des interfaces utilisateur intuitives. En outre, les techniques d'intelligence computationnelle peuvent être utilisées pour permettre un comportement autonome. Dans chacune des deux catégories, une interface homme-machine et un composant de contrôle sont nécessaires pour la gestion minimum du HMS.

II.4.2 Quelques exemples d'application

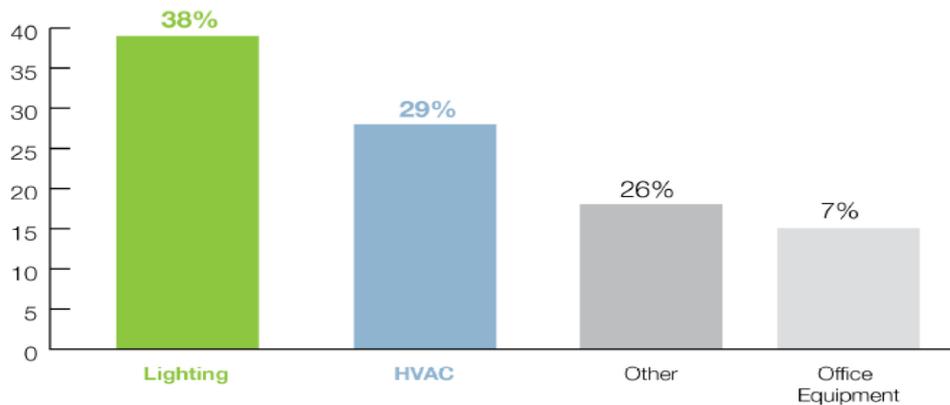


Figure II-4: Consommation annuelle d'énergie dans un bâtiment commercial [10]

La Figure II-4 montre que la majorité de la consommation de l'énergie est principalement partagée entre l'éclairage et le CVC³ [10].

a) Éclairage

Dans le but d'optimiser la consommation d'énergie, un contrôle d'éclairage est intégré dans les systèmes GTE. Ceci permet de fournir une intensité d'éclairage constante de 500 Lux sur les surfaces de travail par le biais d'un capteur de luminosité qui mesure l'intensité lumineuse continuellement. Ainsi, sur la base de la valeur retournée par le capteur et l'intensité lumineuse requise, le contrôleur de lumière active les actionneurs pour ajuster la lumière. D'après les statistiques entre 28% et 66% de l'énergie électrique utilisée pour l'éclairage peut être économisée avec cette méthode de contrôle, selon la saison, la météo et l'emplacement du bâtiment. En outre, il est possible de détecter le niveau d'occupation d'une pièce dans un habitat à l'aide d'un capteur de présence et d'activer l'éclairage pendant de l'occupation. Si par exemple la pièce n'est pas occupée, l'éclairage peut être coupé automatiquement si quelqu'un a oublié de l'éteindre. L'instauration de l'un système de détection de présence peut économiser 13% de l'énergie destinée à l'éclairage. [10]

b) Les unités de CVC

Les unités de CVC dans un habitat sont les appareils les plus gourmandes en termes de consommation d'énergie qui se traduit par des dépenses supplémentaires dans un bâtiment. Ainsi, une mauvaise gestion de l'énergie se répercute négativement sur les coûts pour les produits et services, et par conséquent ceci rend les entreprises moins compétitives sinon ces

³ CVC : Chauffage, Ventilation et Climatisation

dépenses supplémentaires pourraient être investies dans la recherche, le développement de l'entreprise.

En outre, grâce à de nombreux types de capteurs et un système de HEMS, le CVC pourra ajuster ses paramètres en fonction des facteurs suivants : taux d'occupation, la température, la qualité de l'air à l'intérieur, ... Cette opération d'ajustement permettra ainsi une utilisation plus efficace de l'énergie tout en réduisant les coûts opérationnels [10].

c) Contrôle stores

Le contrôle des stores et des fenêtres offre du confort tout en réduisant la consommation d'énergie. Par exemple, les fenêtres peuvent être programmées pour s'ouvrir automatiquement quand le niveau de CO₂ dépasse un certain seuil. Les stores aussi peuvent être ouverts et fermés grâce à un bouton ou programmés pour s'adapter automatiquement à l'intensité de la lumière naturelle [10].

II.5 Types de mécanismes de prélèvement de la consommation d'énergie

II.5.1 Surveillance d'énergie intrusive

Ce mécanisme est semblable aux capteurs de surveillance de l'environnement. Les métriques de mesure d'énergie nécessitent également un protocole sans fil pour communiquer avec le contrôleur. Le prélèvement des mesures de la consommation d'énergie peut être effectué avec deux méthodes : intrusive et non intrusive. Dans l'approche intrusive, la métrique est déployée sur chaque équipement individuel. Un système de surveillance de charge intrusive (ILM) peut être implémenté dans une méthode directe ou indirecte. La méthode directe attache un wattmètre à chaque appareil. Par conséquent, l'ILM direct fonctionne évidemment avec une grande précision mais présente certains inconvénients tels que le coût élevé et l'interruption de l'alimentation pour installer les compteurs. En revanche, la méthode indirecte utilise des capteurs à faible coût pour détecter les signaux mesurables émis par l'appareil pour estimer la consommation électrique. Par exemple, un capteur de lumière peut détecter l'état d'alimentation d'un téléviseur et déduire l'énergie consommée correspondante, un capteur acoustique peut être appliqué pour reconnaître le fonctionnement du compresseur du réfrigérateur [11].

II.5.2 Surveillance d'énergie non intrusive

Contrairement aux approches intrusives, les approches non intrusives reposent sur une seule métrique déployée à l'alimentation principale. Un algorithme NILM⁴ tente d'extraire certaines paternes spécifiques à partir du signal de puissance agrégé pour identifier les équipements correspondants. En supposant qu'il y a N équipements connectés à une seule métrique. Chaque équipement $i \in \{1, \dots, N\}$ peut être dans un état j de m_i modes, qui consomme une puissance de w_{ij} , $j \in \{1, \dots, m_i\}$. Notons $s_{ij}(t)$ comme indicateur booléen de l'état j du dispositif i au temps t :

$$s_{ij} = \begin{cases} 1 & \text{si l'équipement } i \text{ est dans l'état } j \\ 0 & \text{sinon} \end{cases}$$

La consommation d'énergie globale peut être représentée comme suit :

$$x(t) = \sum_{i=1}^N \sum_{j=1}^{m_i} s_{ij}(t) \times w_{ij} + e(t)$$

où $e(t)$: terme de bruit ou d'erreur.

A partir de ce modèle, les vecteurs contenant tous les indicateurs des états $\{s_{ij} \mid i = 1, \dots, N \text{ et } j = 1, \dots, m_i\}$ des équipements de 1 à N peuvent être déterminés en résolvant le problème de minimisation suivant :

$$\hat{s}(t) = \arg \min_{s(t)} \left\| x(t) - \sum_{i=1}^N \sum_{j=1}^{m_i} s_{ij}(t) \times w_{ij} \right\|_d$$

où $\|d$ désigne la distance l_d .

Dans le contexte de NILM, nous pouvons utiliser une certaine distance des métriques telles que l_1 , l_2 . Ce problème est intraitable sur le plan informatique et difficile de le résoudre par des techniques exhaustives sauf si N est petit. Au lieu de cela, des algorithmes heuristiques pourraient être envisagés. Il existe trois principes pour tout algorithme NILM, notamment :

⁴ NILM : Non Intrusive Load Monitoring

- Sélectionnez et caractérisez d'une manière formelle les caractéristiques ou signatures spécifiques.
- Déployez le matériel approprié pour mesurer le signal souhaité.
- Développer un algorithme pour détecter ces caractéristiques / signatures à partir du signal.

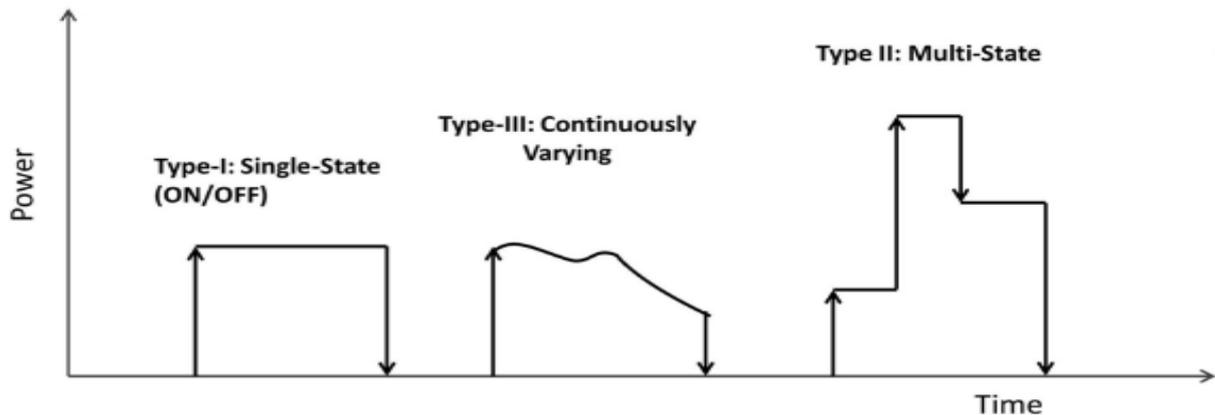


Figure II-5: Différentes catégories de charges basées sur les schémas énergétiques.

La caractérisation des fonctionnalités et des signatures nécessite la connaissance de l'opération des états des appareils, qui peuvent être classés en quatre types comme suit :

- Type I : appareils à deux états (ON/OFF) tels que lampe, grille-pain, etc.
- Type II : machines à états finis avec un nombre fini d'états de fonctionnement, par exemple lavage machine, poêle, etc.
- Type III : appareils à variation continue avec une consommation électrique variable tels que perceuse électrique ou gradateur.
- Type IV : appareils qui restent actifs pendant des jours ou des semaines et consomment un niveau constant de puissance tel qu'un détecteur de fumée, un modem Internet, etc.

La sélection des fonctionnalités/signatures dépend également de la fréquence d'échantillonnage des compteurs et peut être divisée en deux groupes : basse fréquence et haute la fréquence. Les algorithmes basés sur un signal basse fréquence axé sur la détection de l'état stationnaire des fonctionnalités telles que la puissance moyenne, les changements de pas, etc.,

tandis que le matériel haute fréquence nous permet d'identifier les dispositifs de la phase transitoire et des harmoniques [11].

II.5.3 Surveillance de charge hybride

Une approche hybride entre la surveillance de charge intrusive et non intrusive, appelée hybride. Dans cette approche, un réseau de capteurs est déployé dans les habitats pour surveiller les paramètres environnementaux tels que le bruit acoustique, la luminosité, l'occupation, etc., pour détecter le fonctionnement de certains dispositifs. La différence entre les approches hybrides et intrusives est que les capteurs ne sont installés que pour surveiller une partie de tous les appareils tandis que les autres sont toujours identifiés par un algorithme NILM avec mesure de puissance globale. Par conséquent, HLM permet une réduction du coût de l'approche intrusive et une augmentation de la précision de détection du non-intrusif. Dans le cadre de ce projet de fin d'études, nous nous concentrerons sur les méthodes hybrides afin d'améliorer les performances des algorithmes NILM existants [11].

II.6 Conclusion

La Gestion Technique et Énergétique des Bâtiments (HEMS) a été développée dans les années 80 et elle a été considérée comme un axe technologique de développement. En outre, l'avènement de la domotique a permis aujourd'hui de se trouver avec un habitat économe et respectueux de l'environnement. Plusieurs travaux ont été réalisés dans le monde pour illustrer les technologies disponibles afin d'améliorer le mode de vie des personnes, utiliser l'énergie propre et protéger l'environnement.

Dans ce chapitre, on a présenté l'essentiel de ce qui a été fait dans le monde de l'économie de l'énergie et la protection de l'environnement. Il s'est avéré qu'il s'agit d'un problème de disponibilité des technologies pour certifier et assurer la pérennité des systèmes face au grand nombre de services émergents proposés. L'objectif est d'en faire les fondements d'une structure générale d'un habitat, confortable, automatisé et dans lequel le système doit être économe en énergie.

Chapitre III

L'apprentissage automatique

(machine Learning)

Chapitre III

L'apprentissage automatique (Machine Learning)

III.1 Introduction

L'apprentissage automatique est une nouvelle discipline qui consiste à rendre notre environnement de programmation intelligent et permettre la prédiction de l'évolution dans le futur. Cette nouvelle discipline a pu envahir plusieurs domaines d'applications à savoir l'industrie, le comportement des consommateurs, les finances (en détectant les transactions frauduleuses), l'agriculture où les paysans peuvent personnaliser leurs techniques de culture en fonction des conditions météorologiques et les caractéristiques de la portion de leur terrain concernée, etc...

Dans le présent chapitre, nous commencerons par présenter l'apprentissage automatique (Machine Learning) et leurs concepts fondamentaux. Nous présentons également les principales approches de classification en mettant l'accent sur leurs avantages et leurs limitations.

III.2 Machine Learning (ML)

III.2.1 Préambule

Dans cette section, nous présentons un bref historique dans le domaine de l'apprentissage artificiel, (ou Machine Learning, en anglais). Cela nous amène à citer trois grandes âges de l'évolution de l'informatique : le début de l'informatique, de son évolution au cours du temps et au monde d'aujourd'hui et de demain.

Nous pouvons remarquer que l'évolution de l'informatique s'est faite principalement sur deux volets :

- Acquérir la capacité de collecter et de diffuser des informations dans des domaines tels que Data Mining (les fouilles de données), les réseaux et services web, et leurs applications sous Android (Smartphones).

- Gains dans l'intelligence des systèmes informatiques, surtout dans les domaines en relation avec l'intelligence artificielle tels que la vision par ordinateur, etc. [12].

III.2.2 Le paradigme Machine Learning

Dans ce nouveau paradigme, nous donnons plus d'importance aux théories, algorithmes et applications qui sont en relation étroite avec l'intelligence artificielle (IA) là où il y a l'apprentissage. Dans ce contexte, nous constatons que le paradigme de programmation classique a évolué. Dans ce paradigme, il faut prescrire une logique pour faire exécuter une tâche. Cependant, avec l'IA, on programme pour faire exécuter de manière intelligente des tâches répétées nécessitant un raisonnement. Par ce type de démarches, nous serons en mesure de faire doter un programme d'une capacité d'apprentissage [13].

III.2.3 L'histoire du Machine Learning

L'apprentissage artificiel est un nouveau paradigme dans le monde de l'informatique et de l'intelligence artificielle. Il se situe au carrefour d'autres matières telles que les mathématiques, la psychologie, et la logique. C'est en 1920 que les premières expériences deviennent possibles avec cette nouvelle discipline. Dans les années 1960, l'apprentissage artificiel a connu un grand développement surtout avec les approches connexionnistes et la reconnaissance des formes. Les années 1970 ont été connues par l'avènement de nouveaux systèmes impliquant les connaissances et les systèmes experts. Néanmoins, des limites de ces systèmes se font sentir dans les années 1980, pendant lesquelles il y a eu lieu le retour de l'approche connexionniste avec une nouvelle approche d'apprentissage.

Par la suite, les mathématiciens ont commencé à s'éloigner de l'aspect cognitif de l'apprentissage et voir le problème sous forme d'optimisation quand de nouvelles approches ont vu le jour à l'instar des arbres de décision et l'induction des programmes logiques. Par ailleurs, la théorie statistique de l'apprentissage s'est fait sentir dans les années 1990, avec l'apparition de l'ouvrage de Vapnik [13].

III.3 Concepts de l'apprentissage automatique

L'apprentissage chez l'être humain se constitue de plusieurs évolutions qu'il est difficile précisément à les décrire. Les capacités d'apprentissage chez l'être humain lui ont accordé un avantage évolutif déterminant pour son développement. La "capacité d'apprendre" engendre un ensemble d'aptitudes telles que :

- L'acquisition de la capacité de parler en voyant les autres.

- L'acquisition de la capacité de lire, d'écrire, de réaliser des opérations de mathématiques en arithmétiques et logiques avec l'aide d'un tuteur.
- L'acquisition d'habilités motrices et sportives par des exercices. [13]

III.3.1 Qu'est-ce que l'apprentissage artificiel

La capacité d'apprendre des expériences passées et de s'adapter est une caractéristique de la vie supérieure. Il est essentiel que les humains, au tout début de leur vie, apprennent des choses de base comme la reconnaissance vocale, le visage familier, apprendre à comprendre ce qui se dit, marcher et parler.

L'apprentissage automatique est une épreuve de comprendre et de reproduire cette capacité d'apprendre dans des systèmes artificiels. D'un point de vue très schématique, concevoir des algorithmes performants, à partir d'un grand nombre d'exemples (données correspondant à "l'expérience passée"), pour s'adapter à sa nature afin de pouvoir appliquer ce qu'ils ont appris dans des cas futurs.

Nous définissons l'apprentissage automatique comme suit : " Un programme d'ordinateur est capable d'apprendre à partir d'une expérience E et par rapport à un ensemble T de tâches et selon une mesure de performance P , si sa performance à effectuer une tâche de T , mesurée par P , s'améliore avec l'expérience E " [14].

III.3.2 Modélisation

L'apprentissage automatique d'une machine se fait autour d'un ensemble de tâches (T). Ainsi, pour déterminer la performance de la machine, on utilise une mesure de la performance P . La machine peut connaître à l'avance un ensemble d'expériences E et elle va par la suite enrichir cet ensemble après.

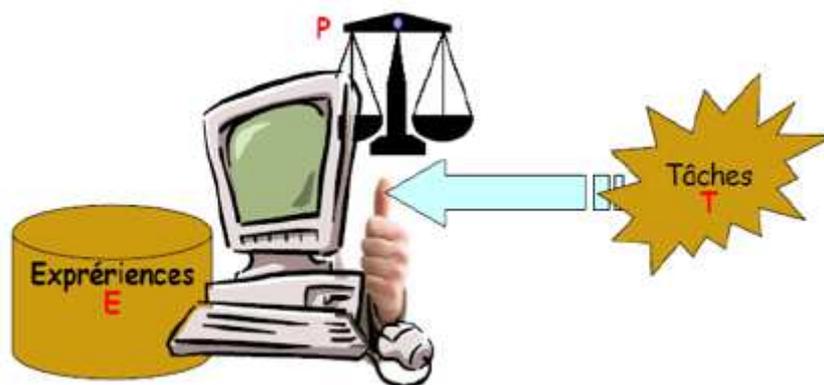


Figure III-1: Schéma de modélisation d'une machine d'apprentissage

Ainsi, l'apprentissage automatique pour la machine est qu'avec l'ensemble des tâches T que la machine doit effectuer, elle utilise l'ensemble des expériences E telles que sa performance sur T soit améliorée [14].

III.3.3 Domaines d'applications de l'apprentissage automatique

Les domaines d'applications de l'apprentissage automatique sont très variés et ils concernent en particulier les domaines qui nécessitent une prise de décision automatisée. Il s'agit, par exemple les activités suivantes [15]:

- Etablissement d'un diagnostic médical à partir de la description clinique d'un patient ;
- Déclenchement d'une alerte en fonction de signaux reçus par des capteurs ;
- La reconnaissance des formes ;
- La reconnaissance de la parole et du texte écrit ;
- Contrôler d'un processus de production et diagnostic des pannes.

III.4 Types d'apprentissage

Les algorithmes d'apprentissage peuvent se catégoriser selon le mode d'apprentissage qu'ils utilisent [13]:

III.4.1 Apprentissage Supervisé

Cette approche permet de concevoir un modèle reliant les données d'apprentissage à un **ensemble de valeurs de sortie** qui explicite un comportement.

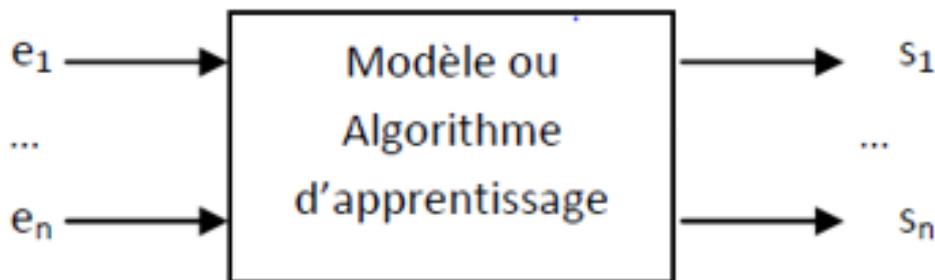


Figure III-2: Schéma d'un modèle supervisé

III.4.2 Apprentissage par renforcement

Les données en entrée sont les mêmes que pour l'apprentissage supervisé, cependant l'apprentissage est guidé par l'environnement sous la forme de récompenses ou de pénalités en fonction de l'erreur commise lors de l'apprentissage.

III.4.3 Apprentissage non supervisé

Il a comme objectif la conception d'un modèle structurant l'information. Dans ce modèle d'apprentissage les comportements ou les classes des données d'apprentissage ne sont pas connus à l'avance, on cherche de les trouver.

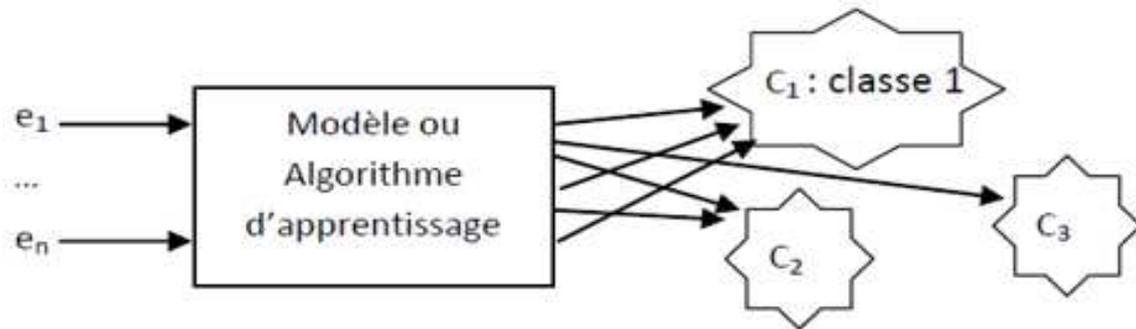


Figure III-3: Schéma d'un modèle non supervisé

III.4.4 Apprentissage semi-supervisé

Les données d'entrée sont composées d'exemples étiquetés et non étiquetés. Cet aspect permet de ne pas en laisser de côté et d'utiliser toute l'information.

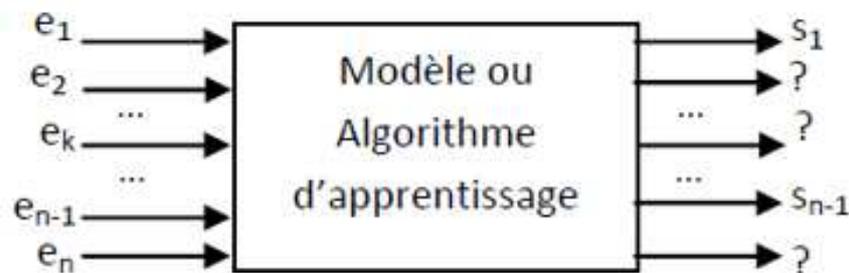


Figure III-4: Schéma d'un modèle semi-supervisé

III.5 Les algorithmes utilisés dans l'apprentissage automatique

Parmi les algorithmes les plus utilisés dans l'apprentissage automatique, nous pourrions citer [15] :

- Les machines à vecteurs support (SMVs)
- Le boosting
- Les réseaux de neurones pour l'apprentissage supervisé ou non-supervisé
- La méthode des k plus proches voisins pour un apprentissage supervisé
- Les arbres de décision

- Les méthodes statistiques : exemple, le modèle de mixture gaussienne
- La régression logistique
- L'analyse discriminante linéaire
- La logique floue
- Les algorithmes génétiques et la programmation génétique

Ces approches peuvent être combinées pour obtenir diverses variantes d'apprentissage. L'utilisation d'une telle approche dépend fortement de la problématique à résoudre.

III.6 Facteurs de pertinence de l'apprentissage automatique

La qualité de l'apprentissage dépend du besoin en amont et de la compétence de l'opérateur pour préparer l'analyse [14]. Elle dépend également de la complexité du modèle et de son adaptation à la problématique traitée. En outre, cette qualité dépendra aussi du mode de mise en évidence des résultats pour l'utilisateur final. Par exemple, un résultat pertinent pourrait être mal mis en évidence par une représentation graphique inadaptée. Avant cela, la qualité du travail dépendra de facteurs initiaux liés à la base de données [14] :

- **Nombre d'exemples** : il y a un compromis entre le nombre d'exemples et la qualité d'analyse. Ainsi, quand le nombre d'exemples est réduit, l'analyse sera difficile alors que lorsqu'il est très grand il a besoin plus de mémoire et l'analyse prendra plus de temps.
- **Nombre et qualité des attributs** : les attributs décrivant les exemples ont un impact sur la qualité. Par exemple, la distance entre deux "exemples" numériques (prix, poids, intensité de bruit, etc.) est facile à établir, cependant celle entre deux attributs catégoriels (couleur, utilité), est plus délicate.
- **"Bruit"** : Le nombre et la localisation des valeurs douteuses ou non conformes au modèle de distribution générale des "exemples" sur leur espace de distribution

III.7 Différentes étapes de l'apprentissage automatique

La machine d'apprentissage ayant comme tâche est d'apprendre une fonction à travers des exemples (exemple une fonction de prise de décision) ne permet pas de nous indiquer comment obtenir un classificateur adapté à la tâche considérée [15]. Dans ce qui suit, nous présentons les différentes étapes que l'approche Machine Learning recommande pour atteindre un tel objectif.

III.7.1 Ensemble de données

Lorsque nous construisons un modèle qui réduit les risques expérimentaux calculés sur les données d'apprentissage, l'erreur d'apprentissage obtenue ne peut pas être considérée comme une bonne mesure d'une erreur de généralisation : elle est considérée comme biaisée.

Pour obtenir un estimé non biaisé de l'erreur de généralisation, il est important de mesurer l'erreur dans des exemples qui n'ont pas été utilisés pour entraîner le modèle. Pour ce faire, nous divisons l'ensemble de données disponibles en deux parties : un sous-ensemble d'apprentissage, dont les données seront utilisées pour entraîner le modèle, et un sous-ensemble de tests, dont les données ne seront utilisées que pour évaluer les performances du modèle entraîné. Cela donne une erreur de test qui est un estimé bruité mais non biaisée de l'erreur de généralisation.

De plus, le même ensemble d'apprentissage est souvent partagé entre un sous-ensemble qui est utilisé pour apprendre les paramètres du même modèle, et un autre sous-ensemble appelé "de validation", qui est utilisé pour la sélection du modèle. Les différentes opérations qui doivent être effectuées avant de soumettre des données à l'algorithme d'apprentissage sont :

1) Acquisition des données

Si les données proviennent d'une source analogique, elles doivent d'abord être converties pour obtenir une représentation pouvant être traitée par un programme informatique.

2) Prétraitement

L'application de la phase de prétraitements sur les données brutes peut voir deux apports : Réduire la taille de l'information qui va être présentée au classificateur, ce qui se traduit par un gain en temps. Éliminer l'information non pertinente qui peut être une source de confusion pour le classificateur. Nous allons présenter quelques méthodes utilisées lors de la phase de prétraitements.

Les bons résultats qu'un classificateur automatique peut fournir dépendent en grande partie de l'étape de prétraitement. Les données d'un mauvais prétraitement mettraient en péril la qualité du classificateur. Cette étape consiste en une suite de traitements sur données brutes pour extraire des informations et ne conserver que celles qui sont utiles pour la classification. L'application de l'étape de prétraitement sur les données brutes peut présenter deux avantages : la réduction de la quantité d'informations qui seront fournies au classificateur, ce qui permet un gain de temps ; l'élimination des informations non pertinentes qui peuvent influencer le travail

du classificateur. Dans ce qui suit, nous présenterons certaines méthodes utilisées pendant la phase de prétraitement.

- **Extractions des caractéristiques :** Les techniques d'extraction des caractéristiques éliminent les informations redondantes et ne conservent que les informations pertinentes. Nous pouvons, par exemple, utiliser des techniques de réduction telles que la décomposition en ondelettes et l'analyse en composantes principales (ACP) pour compresser les données tout en conservant des informations aussi utiles que possible.
- **Sélections des caractéristiques :** Dans certains scénarios, après l'application de la technologie de compression des données, le nombre de propriétés reste quand même toujours élevé pour servir comme informations d'entrée pour un classificateur. De plus, la réduction ne permet pas nécessairement de choisir des propriétés permettant une bonne distinction entre éléments différents. Certaines peuvent causer une confusion lors de l'affectation des éléments à des classes. On dit qu'une caractéristique est pertinente si elle contribue fortement à une bonne distinction entre deux éléments appartenant à deux groupes hétérogènes et à un bon rapprochement des éléments d'un même groupe. Les méthodes de sélection des caractéristiques essaient de sélectionner un sous-ensemble de caractéristiques appropriées parmi celles obtenues durant la phase d'extraction. Cette sélection vise deux objectifs : surmonter les limitations techniques : aucun classificateur ne peut fonctionner correctement avec plus de cinquante propriétés ; permettre un apprentissage accéléré : moins il y a de propriétés, plus les processus de calcul et d'apprentissage sont rapides.

3) Conversion

Il s'agit de la conversion des données au format spécifié par l'algorithme utilisé lors de la phase de classification. La représentation vectorielle est très répandue, parce qu'elle permet de concevoir des algorithmes relativement indépendants du type d'objet à classer. Dans ce format, les données sont représentées sous forme de vecteurs où chacune des composantes représente une caractéristique de l'objet. La plupart des algorithmes de classification ont du mal à gérer de des vecteurs de grande dimension.

4) Post-traitement

Dans certains scénarios, nous devons normaliser les données dans le format d'entrée. Par exemple, dans le cas d'un réseau de neurones, nous conseillons que la moyenne de chaque composante du vecteur sur l'ensemble des exemples d'entraînement soit proche de zéro.

III.7.2 Entraînement ou apprentissage

La phase d'entraînement permet la sélection d'une fonction $f \in F_{\gamma}$ (à trouver une évaluation des paramètres γ_i). La sélection des paramètres γ_i est faite par un algorithme d'apprentissage qui reçoit en entrée l'ensemble d'apprentissage et un ensemble de paramètres d'apprentissage : $\{p_1, p_2, \dots, p_k\}$. Ainsi, les données de l'ensemble d'apprentissage induisent l'apprentissage et l'ensemble des paramètres γ_i résultant de l'apprentissage représente le modèle. Une machine d'apprentissage munie d'un modèle est appelée machine entraînée [15]. La Figure III-5 illustre le processus d'un schéma d'un tel apprentissage.

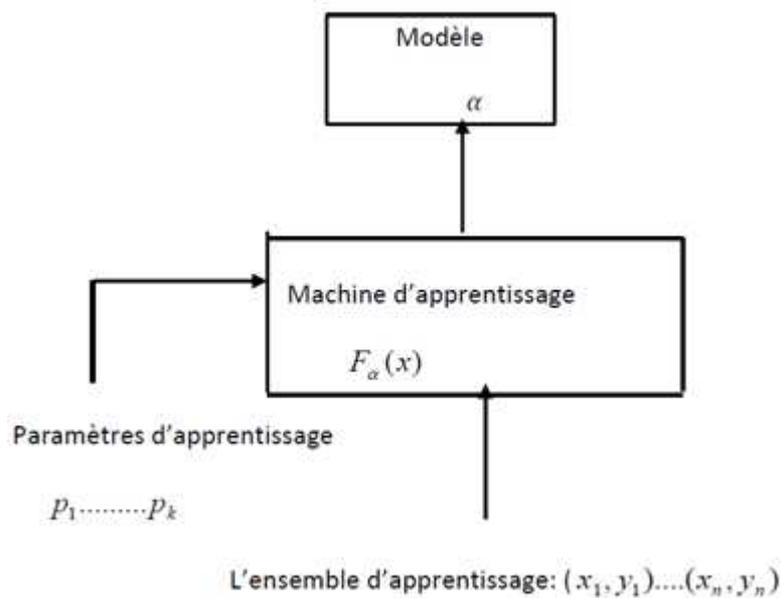


Figure III-5: Entraînement d'une machine d'apprentissage

III.7.3 Evaluation du modèle

Une fois le modèle obtenu, il est intéressant d'évaluer ses performances sur un ensemble indépendant de données : le test set. En effet, on ne peut se fier aux résultats obtenus sur l'ensemble d'apprentissage car la machine d'apprentissage a perdu son indépendance face à ces données. Cette phase permet de se rendre compte du pouvoir de généralisation du classificateur, c'est à dire sa capacité à obtenir de bons résultats sur n'importe quel ensemble de données provenant de la même distribution. Dans les cas où l'on dispose de trop peu de données, on peut utiliser une technique de *validation croisée*, pour générer plusieurs paires de sous-ensembles entraînement / test. Cette technique est coûteuse en temps de calcul, mais permet d'obtenir un bon estimé de l'erreur de test, tout en conservant suffisamment d'exemples pour l'entraînement.

Une fois le modèle est obtenu, il faut bien évaluer ses performances sur un ensemble de données indépendant (l'ensemble des tests). En fait, nous ne pouvons pas faire confiance aux résultats obtenus sur l'ensemble d'apprentissage car la machine d'apprentissage n'a pas son indépendance face à ces données. Cette phase permet de se rendre compte que le classificateur n'a pas la capacité pour obtenir de bons résultats sur n'importe quel ensemble de données de la même distribution. Ainsi, dans les cas où nous disposons de peu de données, nous pouvons utiliser la technique de validation croisée pour créer plusieurs paires de sous-ensembles "entraînement/test". Cette technique est coûteuse en termes de temps de calcul mais elle permet d'obtenir une bonne estimation de l'erreur de test, tout en conservant suffisamment d'exemples pour l'entraînement. [16]

Validation croisée

C'est une technique qui permet de tester un modèle d'apprentissage. La validation croisée se décline en plusieurs sous-méthodes. La validation croisée est disponible de plusieurs manières. Néanmoins, la méthode la plus répandue est celle "k-Fold" avec $k \in [4,10]$. Si nous avons d'une base d'apprentissage A_p qui contient p éléments, cette dernière se présente comme suit :

$$A_p = \{x_1, x_2, \dots, x_p\}$$

La validation croisée consiste à appliquer les cinq étapes suivantes :

1. Découper l'ensemble des exemples en k sous-ensembles disjoints de même cardinalité (le nombre d'éléments par sous-ensemble est : p/k).
2. Lancer le processus d'apprentissage sur $(k-1)$ sous-ensembles parmi les k sous-ensemble.
3. Calculer l'erreur sur le $k^{\text{ième}}$ sous-ensemble qui n'était pas pris lors de la $2^{\text{ième}}$ étape..
4. Répéter le processus p fois pour engendrer l'erreur.
5. Calculer la moyenne des k erreurs précédentes (erreur moyenne).

La validation croisée permet d'obtenir une estimation de l'erreur de généralisation et d'éviter le sur-apprentissage.

III.7.4 Exploitation

Un classificateur correspond bien à une machine entraînée. Lorsque nous disposons d'un classificateur efficace pour une tâche particulière, nous pouvons l'utiliser pour faire des

prédictions sur de nouveaux exemples. Les entreprises qui vendent ce type de classificateurs, entraînent souvent la machine d'apprentissage sur un corpus relatif aux catégories définies par le client. De cette façon, le client ne reçoit qu'une machine entraînée sans se soucier comment se fait l'entraînement et les paramètres utilisés. Par exemple l'application de la reconnaissance vocale (Babel Speech) destinée au grand public propose un classificateur entraîné. Il existe souvent une étape d'adaptation (audio ou écriture) visant à déterminer le modèle le plus approprié pour l'utilisateur.

III.8 Quelques méthodes de classification Automatique (Clustering)

III.8.1 La méthode K-Means

L'approche K-Means a été mise au point par McQueen en 1967 [17]. Elle se base sur un algorithme d'apprentissage non supervisé. Il permet d'associer chaque point au cluster dont le centroid est le plus proche où les coordonnées de ce dernier sont la moyenne arithmétique de tous les points dans le cluster c'est à dire chaque cluster est représenté par son barycentre [16].

L'idée principale dans K-means est de définir les K centroids arbitraires C_1, C_2, \dots, C_k où K représente le nombre de clusters fixé a priori. Chaque C_i représente le centre d'un cluster. Ces centroids doivent être placés dans des positions différentes et le meilleur choix est de les placer le plus possible éloignés les uns des autres. L'étape suivante dans K-means consiste à prendre chaque point appartenant à l'ensemble de données et le joindre au plus proche centroid c'est-à-dire chaque classe S_i (cluster) sera représentée par un ensemble d'individus les plus proches de leur C_i . Lorsque tous les points sont affectés à leurs clusters correspondants, la première étape est achevée et un groupage préliminaire est fait. À cette étape, nous avons besoin de recalculer les K nouveaux centroids m_i des clusters issus de l'étape précédente qui vont remplacer les C_i (m_j est le barycentre du cluster S_j , calculé en utilisant les nouveaux clusters obtenus). Après, on répète le processus de clustering jusqu'à atteindre un état stable où aucune amélioration n'est possible. Nous pouvons constater au cours de l'exécution de K-means que les K centroids changent leurs positions après chaque étape jusqu'à plus de changements sont possibles.

III.8.2 Méthodes hiérarchiques

Dans [18] [19], les auteurs ont donné le premier fondement des méthodes hiérarchiques. Ce nouveau type de clustering consiste à réaliser une suite de regroupement en clusters de moins

en moins fines en assemblant à chaque étape les objets (simple élément) ou les groupes d'objets (un cluster) les plus proches. Ce qui permet de générer une arborescence de clusters. Cette méthode de clustering implique la mesure de similarité dans le processus de regroupement pour illustrer l'homogénéité ou l'hétérogénéité des classes.

Dans les méthodes hiérarchiques, au début chaque individu forme une classe c'est-à-dire au début on commence avec un nombre de n classes qu'on cherche à le réduire ($n_{\text{new}} < n$). Le processus de regroupement est exécuté d'une manière itérative de sorte que dans chaque étape deux classes sont fusionnées ensemble où ces deux classes sont les plus "proches" en fonction de leur dissimilarité ou bien ajouter un nouveau élément à une classe où cet élément est plus proche de cette classe que de toutes les autres. La dissimilarité est caractérisée par une valeur appelée aussi indice d'agrégation.

On distingue plusieurs méthodes qui sont basées sur l'approche hiérarchique et parmi ces méthodes on cite : la classification ascendante hiérarchique (CHA) qui est la plus répandue. Dans cette méthode, le mot "ascendant" veut dire qu'au début chaque individu représente un cluster, puis on cherche à les rassembler en clusters de plus en plus grands. En outre, le qualificatif "hiérarchique" désigne qu'une hiérarchie sera générée par la suite. Cette approche a été améliorée par en 2002 par P. Bertrand (Classification Ascendante 2-3 Hiérarchique) [20].

III.9 Introduction au Deep Learning

Depuis 2006, le domaine de l'apprentissage automatique a fait de grandes avancées avec l'apprentissage de réseaux profonds (Deep Learning). Cette nouvelle approche permet, à partir de données d'entrée, d'en extraire une représentation plus riche. Les réseaux profonds sont composés de plusieurs couches de neurones où chaque couche représente une étape où les données sont représentées de façon un peu plus complexe (abstraite) en se basant sur ce qui a été appris dans la couche précédente.

Deep Learning est à l'image de l'apprentissage humain qui commence par apprendre des concepts simples, comme l'addition et la soustraction en mathématiques, pour ensuite se baser sur ces concepts afin d'en apprendre des plus complexes, comme la multiplication et le principe de fonction. [21]

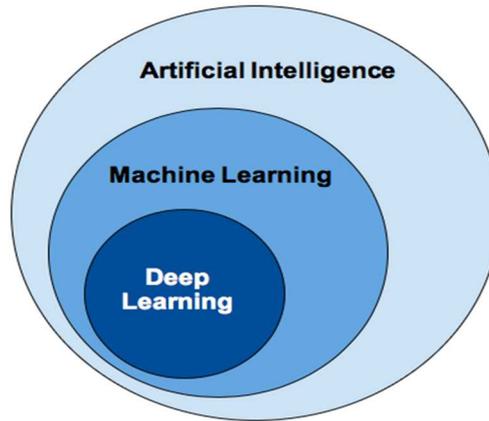


Figure III-6: La relation entre l'intelligence artificielle, le ML et le deep Learning

III.9.1 Définition de l'apprentissage en profondeur

L'apprentissage en profondeur (Deep Learning : DL) est une branche de l'apprentissage automatique qui permet d'enseigner aux ordinateurs à faire ce qui vient naturellement aux humains : apprendre à partir de l'expérience. Les algorithmes d'apprentissage automatique impliquent des méthodes de calcul pour "apprendre" des informations directement à partir de données sans se baser sur une équation prédéfinie comme modèle.

DL est considéré comme une classe de techniques d'apprentissage automatique qui exploitent de nombreuses couches de traitement d'informations non linéaires pour l'extraction et la transmutation de fonctions supervisées ou non supervisées, et pour l'analyse et la classification de modèles. Les caractéristiques et les concepts de niveau supérieur sont donc définis en termes de niveaux inférieurs, et une telle hiérarchie de caractéristiques est appelée architecture profonde. La majorité de ces modèles sont fondés sur un apprentissage non supervisé des représentations [22].

III.9.2 Domaines d'application de deep learning

L'apprentissage en profondeur investit progressivement dans notre quotidien, nous citons dans ce qui suit, quelques domaines d'applications du deep learning :

- La reconnaissance vocale
- Le tagging automatique des morceaux de musique
- La synthèse vocale d'une manière avancée
- L'étiquetage automatique d'images et la reconnaissance des formes
- La conception de nouvelles molécules pharmaceutiques

- Dé-bruitage des images (restauration des images dégradées).

Toutes les applications qui demandent de l'apprentissage et de l'intelligence sont mises en œuvre aujourd'hui en utilisant les techniques de Deep Learning [23].

III.9.3 Différents types de modèles de deep learning

Il existe plusieurs variantes d'architectures basées sur le deep learning. La majorité d'entre elles sont issues des premières architectures parentales. Par ailleurs, c'est un peu difficile de comparer les performances de ces architectures, car elles ne sont pas toutes évaluées sur le même benchmark (les mêmes ensembles de données). Deep Learning est en croissance continue et rapide, de nouvelles architectures et algorithmes voient le jour presque chaque semaine [24].

a) Réseaux de neurones convolutionnels (CNN)

Les réseaux de neurones convolutifs (CNN : Convolutional Neural Network) sont un type de réseaux de neurones artificiels. La première version des CNNs a été créée en 1989 par LeCun et appliquée à la reconnaissance d'écriture manuscrite [25].

Les CNNs sont des structures de réseaux de neurones particulières dans lesquelles l'opération élémentaire dans ce type de réseaux est une convolution et non pas une multiplication matricielle. Ces réseaux ont été proposés pour exploiter les données ayant une structure spatiale ou temporelle à savoir les images, les vidéos. Par exemple, pour les images, il est usuel d'utiliser une entrée de taille $256 \times 256 \times 3$ pour représenter une image classique sur Internet. Avant les CNNs, des grandeurs et/ou attributs étaient calculés sur les images pour servir d'entrées au réseau de neurones. Néanmoins, si l'entrée est une image donc ça va être une liste de presque 200000 entiers pour la représenter. Ainsi, l'utilisation de couches complètement connectées serait une perte de temps de calcul et de poids à apprendre. Dans CNNs, les couches de convolution permettent de mieux engendrer la distribution spatiale grâce à une sensibilité locale tout en essayant de maintenir un nombre de poids à estimer très raisonnable.

Le fonctionnement des CNNs est simple et fondé sur une convolution. La sortie est obtenue par une opération de convolution sur l'image d'entrée et un nombre fixé de filtres. En outre, la sortie d'un filtre est le résultat de la multiplication du contenu d'une fenêtre glissante appliquée à l'image par le filtre.

b) Deep Belief Network (DBN)

DBN est une architecture des réseaux de neurones utilisée pour mettre en œuvre le modèle génératif de Hinton. Cette architecture contient deux parties [26]. La première partie est composée d'un ensemble de couches de compression qui ont comme rôle la conversion des données "entrées X " en une représentation abstraite " $X_{abstrait}$ " tandis que la seconde partie permet de convertir cette représentation en labels de classification y .

Dans la première partie, il a été visé d'apprendre la distribution des données X présentées en entrée sans tenir compte des labels y . Cette partie est composée d'une succession de couches dont chacune comportera une représentation plus abstraite ou compressée que la précédente. Par ailleurs, Hinton et al. ont proposé un algorithme rapide pour la phase d'apprentissage (entraînement). Ainsi, une fois cette phase est accomplie, la première section du réseau de neurones comportera une représentation hiérarchique des données en entrée et la dernière couche encodant la représentation $X_{abstrait}$ la plus abstraite. Cette première phase peut être vue comme une initialisation efficace du DBN (c'est une phase de pré-entraînement).

En outre, le rôle de la seconde partie de l'architecture DBN est de permettre la conversion de la représentation abstraite " $X_{abstrait}$ " en labels y utilisables par exemple dans le contexte d'un apprentissage supervisé. Dans l'exemple des digits, cette représentation sera formée d'une couche de sortie de dix neurones (un neurone par digit).

L'entraînement du modèle DBN est considéré comme accompli lorsque la performance de ce dernier évaluée sur un ensemble de validation distinct de l'ensemble d'entraînement ne s'améliore plus d'une manière significative. Cette seconde phase est appelée également "fine-tuning" ; elle est généralement plus lente que la première phase d'initialisation.

c) Auto-encodeurs (Autoencoders)

L'autoencodeur est un modèle permettant de réaliser une compression de l'entrée (encodeur) et une décompression de celle-ci (décodeur) [20]. Il vise à compresser l'entrée tout en gardant l'erreur $\|X - X'\|^2$ très faible. Dans ce modèle, le poids du décodeur est lié au poids de l'encodeur.

Architecture

L'autoencodeur est un type particulier du DNN (Deep Neural Networks) sans classe étiquetée (sans labels), dont les vecteurs de sortie ont la même dimension que les vecteurs

d'entrées. Il est généralement utilisé dans l'encodage des données. Un auto-encodeur a une couche d'entrée (couche L_1) qui représente les vecteurs de données ou de caractéristiques, une ou plusieurs couches cachées qui représentent la caractéristique transformée (couche L_2) et une couche de sortie (couche L_3). Lorsque le nombre de couches cachées est supérieur à un, l'auto encodeur est considéré comme profond. La dimension des couches peut être soit plus petite lorsque la compression est visée ou grande lorsqu'on vise d'augmenter la dimension d'espace [27].

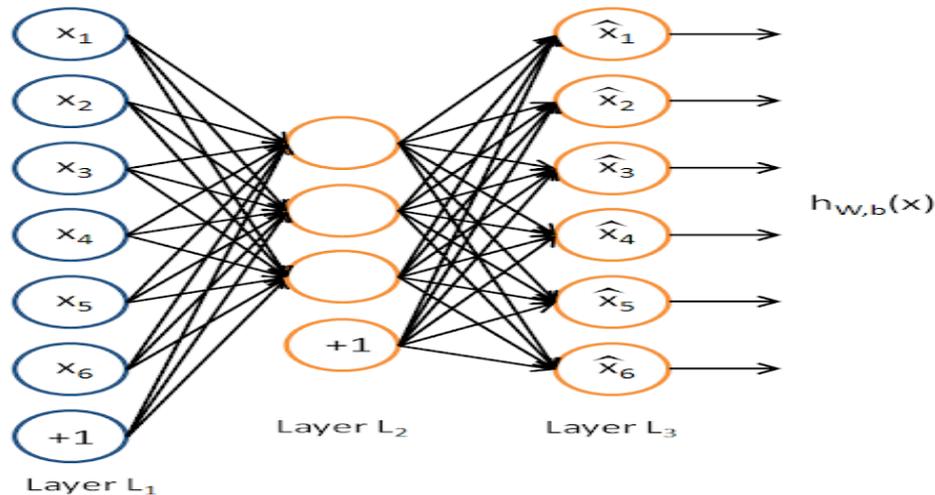


Figure III-7: L'architecture des Auto-encodeurs

Entraînement d'auto-encodeurs

- Auto-encodeur est entraîné d'une manière non supervisée pour extraire une représentation compacte.
- Entraînement d'une couche à la fois.
 - Entraînement de la couche la plus externe.
 - Ajout d'une nouvelle couche qui a été entraînée individuellement.
- La non-linéarité du transfert entre les couches.
 - Essentiel, sinon plusieurs couches non linéaires peuvent être simplifiées en une seule couche.
 - Apprentissage des poids par descente du gradient avec une possibilité d'utiliser la méthode ACP.
- La couche de sortie sera ajoutée à l'encodeur, avec entraînement supervisé, ce qui permet :
 - Un entraînement complet de la couche de sortie par "rétropropagation" et l'ajustement des poids de l'encodeur.

Clustering intégré en profondeur (DEC)

Les encodeurs automatiques permettent de découvrir les clusters qui ont des instances similaires dans un ensemble de données non étiqueté. Autrement dit, sur la base d'un ensemble de données et du nombre de clusters (K), nous devons trouver une représentation de faible dimension dans laquelle il y a les K clusters.

DEC a été proposé par Xie et al. (2016). Il est considéré parmi les premiers modèles qui ont utilisé des auto-encodeurs profonds pour le clustering. Il s'exécute en deux phases. Dans la première phase, l'auto-encodeur est initialisé et entraîné de la manière habituelle (sans clustering) et dans la deuxième phase, le décodeur est impliqué et l'encodeur est utilisé d'une manière précise pour produire de meilleurs clusters avec une procédure de "durcissement de clusters" [28].

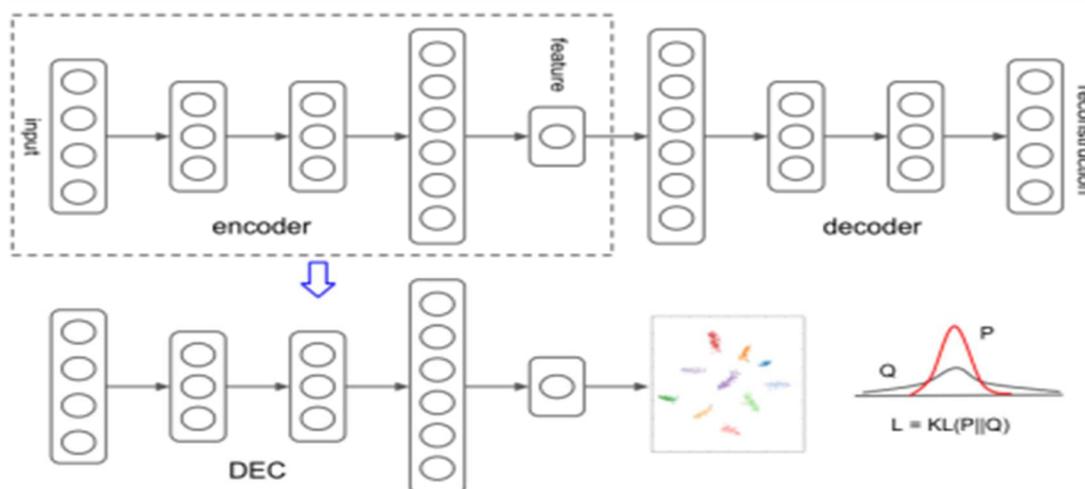


Figure III-8 : Diagramme du modèle DEC

Dans ce qui suit, nous détaillons un peu plus la deuxième phase. Après avoir entraîné l'auto-encodeur, l'approche K-means sera exécutée sur la couche cachée pour obtenir les centroids initiaux $\{\mu_i\}_{i=1}^K$. L'hypothèse est que les affectations initiales des clusters sont généralement correctes, mais il nécessite d'être affiner pour qu'elles soient plus différentes et séparées.

Ainsi, tout d'abord, on fait associer chaque point caché aux centroids $\{\mu_i\}_{i=1}^K$ des clusters en faisant appel à la distribution T de Student :

$$q_{ij} = \frac{\left(1 + \frac{\|z_i - \mu_j\|^2}{\alpha}\right)^{-\frac{\alpha+1}{2}}}{\sum_{j'} \left(1 + \frac{\|z_i - \mu_{j'}\|^2}{\alpha}\right)^{-\frac{\alpha+1}{2}}}$$

Les auteurs ont fixé ($\alpha = 1$) (les degrés de liberté), de sorte que ce qui précède peut-être simplifié et représenté par l'équation ci-dessous :

$$q_{ij} = \frac{\left(1 + \|z_i - \mu_j\|^2\right)^{-1}}{\sum_{j'} \left(1 + \|z_i - \mu_{j'}\|^2\right)^{-1}}$$

où $f_i = \sum_j q_{ij}$ est la fréquence du cluster soft du cluster j . Intuitivement, la quadrature q_{ij} rapproche la distribution de probabilité des centroïdes.

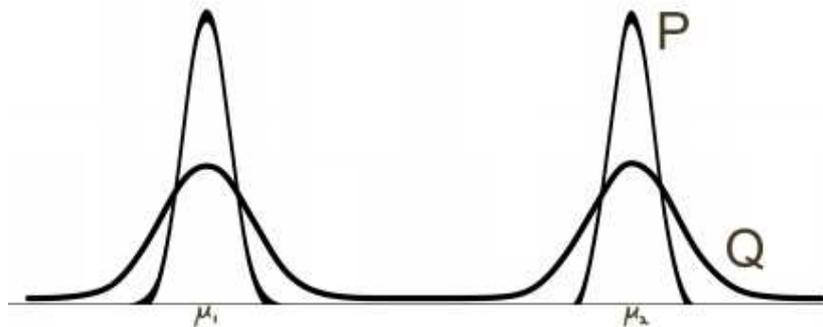


Figure III-9 : La distribution auxiliaire P est dérivée de Q , mais plus concentrée autour des centroïdes.

Enfin, on définit l'objectif qui permet de minimiser la divergence KL entre la distribution d'affectation souple Q et la distribution auxiliaire P :

$$L = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

L'utilisation de la rétropropagation standard et la descente de gradient stochastique permet de former l'auto-encodeur pour produire des points latents afin de minimiser la divergence KL. On répète ainsi le processus jusqu'à ce que les affectations des clusters soient stables [28].

III.8 Conclusion

Dans ce chapitre, nous avons présenté les méthodes de classification automatique et les différents domaines sous-jacents à ces méthodes. Cette étude nous a permis de savoir les avantages et les limitations de chacune des méthodes présentées. Nous avons conclu que le classificateur pourra être une source d'erreurs si ce dernier est de mauvaise qualité. Ainsi, le choix d'un classificateur efficace est essentiel pour garantir une classification avec un taux d'erreurs minimale.

Dans le chapitre suivant, nous illustrons l'utilisation d'une technique de classification automatique à travers une application permettant la détection des anomalies dans la consommation d'électricité.

Chapitre IV

Application : Classification automatique des évènements

Chapitre IV

Application : Classification automatique des évènements

IV.1 Introduction

Dans ce chapitre, nous présentons notre application qui consiste à détecter les états des équipements en traitant seulement la charge électrique agrégée. Pour cela nous avons utilisé la méthode d'apprentissage approfondi (DEC). Cette méthode de classification automatique (clustering) se base sur un apprentissage approfondi non supervisé. Nous avons pris comme environnement de test une maison surveillée dans lequel il y a un ensemble d'équipements tels qu'une machine à laver, un lave-vaisselle, un chauffage et un climatiseur, etc. Par ailleurs, le but de cette application est de créer des compteurs intelligents (soft sensors) qui permettent de connaître l'équipement qui est actif à un instant donné en fonction de la consommation de l'électricité.

Dans ce que suit, nous présentons l'environnement de travail et les outils logiciels utilisés. Nous présentons également les opérations de collection, de prétraitement des mesures, de modélisation et de classification des évènements et enfin l'évaluation de notre solution proposée.

IV.2 Environnement du développement

Afin de réaliser ce projet, nous avons combiné entre différents outils logiciels open-source dédiés à l'analyse des données et l'apprentissage approfondi. Nous notons que la partie modélisation a été développée en utilisant le langage Python.

IV.2.1 La plateforme KNIME

KNIME (Konstanz Information Miner) est un logiciel open-source permettant l'analyse des données en utilisant une interface graphique analogue à LabVIEW. Il permet d'intégrer plusieurs langages de programmation et outils ainsi que la création automatique de résultats. KNIME contient un ensemble d'outils pour l'apprentissage automatique (Machine Learning) et l'exploration de données en utilisant une interface de workflow modulaire. Il dispose d'une

interface utilisateur graphique permettant la construction de workflow par l'assemblage de bout à bout les nœuds où chacun d'eux réalise une opération spécifique telle que le formatage des données, leur modélisation, l'analyse et l'affichage des résultats.

L'avantage de KNIME est qu'elle permet d'utiliser un certain nombre d'algorithmes avec peu de connaissances en programmation et la possibilité d'intégrer des nœuds Python ce qui nous facilite la combinaison entre les workflows KNIME et Python.

KNIME est structurée en une architecture modulaire et extensible, ainsi un certain nombre de nœuds sont disponibles par la communauté. Il est également possible de designer ses propres nœuds KNIME en Java (Figure IV-1).

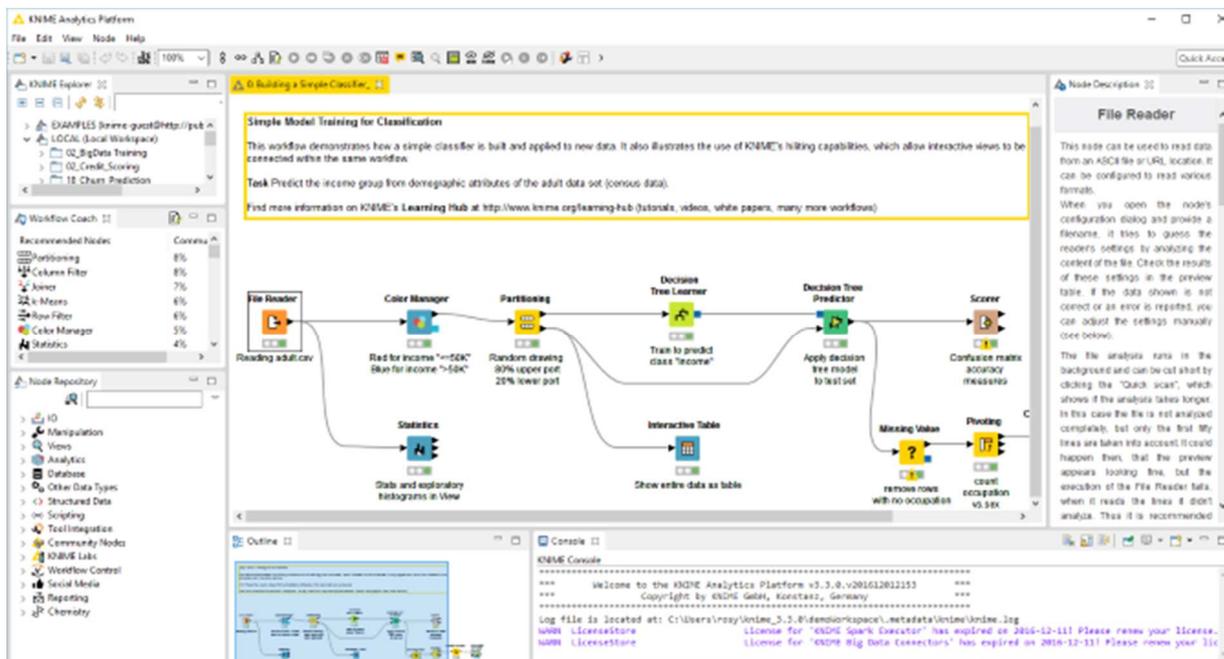


Figure IV-1: Interface de la plateforme KNIME

IV.2.2 Python

Python est langage de programmation multi-paradigme qui a été créé en 1989 par *Guido van Rossum*. Son nom est venu d'un hommage à la série télévisée *Monty Python's Flying Circus*. La première version de ce langage a été publiée en 1991 et sa dernière version 3.7 a été publiée en juin 2018.

Python présente plusieurs caractéristiques intéressantes :

- Python est gratuit et multiplateforme. : Windows, Mac OS X, Linux, Android, iOS.

- Il demande peu de connaissance sur le fonctionnement d'un ordinateur pour qu'il soit utilisé.
- C'est un langage interprété où un script Python n'a pas besoin d'être compilé tout entier pour être exécuté.
- C'est un langage extensible Il permet de favoriser la programmation structurée fonctionnelle et orientée objet.
- Il est relativement facile à le prendre en main.
- Python dispose d'une syntaxe très simple et combinée à des types de données évolués.
- Il a la possibilité de gérer ses ressources (mémoire, descripteurs de fichiers...) sans l'implication du programmeur.
- Il est très utilisé en l'intelligence artificielle et en analyse de données.

Toutes ces caractéristiques ont fait de Python, un outil très adapté pour implémenter notre application

IV.2.3 La bibliothèque TensorFlow

La bibliothèque TensorFlow représente un framework de programmation pour le calcul numérique de Google développé par l'équipe "Google Brain" spécialisé dans l'intelligence artificielle. Ce framework est devenu open-source en Novembre 2015, et l'un des frameworks les plus utilisés pour l'apprentissage approfondi. Les principales caractéristiques de tensorflow sont les suivantes :

- Multi-plateformes : Windows, Linux, Mac OS, et même Android et iOS.
- APIs en Python, C++, Java et Go. Néanmoins l'API Python est la plus complète.
- Temps de compilation est très court grâce au backend en C/C++.
- Supporte les calculs sur CPU, GPU et même le calcul distribué sur cluster.
- Une documentation bien fournie avec plusieurs exemples et tutoriels
- Last but not least : vu que le Framework vient de Google et que ce dernier ait migré la quasi-totalité de ses projets liés au Deep Learning en TensorFlow, cela semble rassurant. Cependant, dans le cadre de notre PFE, nous utilisons cette plateforme via la bibliothèque Keras.

IV.2.4 La bibliothèque keras

C'est une API de haut niveau qui permet de créer des modèles d'apprentissage en profondeur. Elle est capable de s'exécuter sur Tensorflow et utilisé pour le prototypage rapide. Les modèles Keras sont construits en reliant des blocs configurables ensemble. Ainsi, cette bibliothèque nous permet de générer de nouvelles couches et des fonctions. Elle permet également de développer de nouveaux modèles avec peu de restrictions et un petit nombre de lignes de code.

IV.2.5 La bibliothèque Scikit-Learn

Scikit-learn est une bibliothèque libre de Python qui est conçue pour l'apprentissage automatique. Elle est le résultat des efforts de plusieurs contributeurs notamment dans le monde académique à l'instar de l'INRIA. Elle contient des fonctions pour estimer des forêts aléatoires, des algorithmes de classification, des régressions logistiques, et machines à vecteurs de support (SVMs). Elle est développée pour qu'elle cohabite avec d'autres bibliothèques libres Python, telles que Numpy et SciPy. Scikit-learn dispose d'une interface cohérente avec les algorithmes d'apprentissage automatique actuels, facilitant ainsi l'intégration de la machine learning dans les systèmes. Elle constitue un standard de l'industrie pour l'apprentissage automatique avec le langage Python.

IV.2.6 La bibliothèque Numpy

NumPy représente le paquet prépondérant du calcul scientifique avec Python. Il contient les éléments suivants :

- Un objet tableau à N dimensions qui est puissant
- Fonctions très développées
- Il dispose d'outils permettant l'intégration du code en C / C ++
- Il est capable de réaliser des opérations en : algèbre linéaire, de transformée de Fourier et de générer des nombres aléatoires.

La bibliothèque NumPy peut également être utilisée comme conteneur multidimensionnel de données génériques et par le biais de cette bibliothèque des types de données arbitraires peuvent être définis. Cela permet à NumPy d'intégrer de manière transparente avec une variété de bases de données. Elle est sous licence BSD, d'où sa réutilisation avec peu de restrictions est possible.

IV.3 Collection des données

Pour cette étape, il existe des différentes bases de données (Dataset) qui ont été conçues pour adresser le problème de NILM. Ces Datasets peuvent être catégorisées en fonction de la fréquence de prélèvement de la donnée. Les datasets avec une fréquence de prélèvement ≤ 300 record par seconde sont considérées comme *low sampling frequency datasets* et les datasets avec une fréquence de prélèvement ≥ 300 record par seconde sont considérées comme *high sampling frequency datasets*.

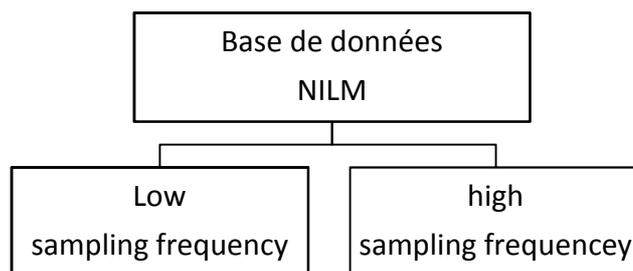


Figure IV-0-2: Base de données NILM

Comme on est intéressé par la classification automatique des évènements, les datasets à faible fréquence de prélèvement peuvent avoir une lecture insuffisante de l'évènement (perte de données). A cet effet, on a choisi de travailler avec *BLUED dataset* qui est une base de données (dataset) à haute fréquence de prélèvement.

IV.3.1 BLUED Dataset

Le dataset BLUED contient des données haute fréquence (12 kHz=12000 records par seconde) de courant et de tension bruts de toute la maison durant une semaine et la puissance active calculée correspondante (60 Hz). L'ensemble de données contient également une liste d'horodatages d'évènements. Autrement dit, chaque fois qu'un état de consommation d'énergie d'un appareil change de 30 watts ou plus et dure au moins 5 secondes. Le nombre d'appareils est d'environ 50. Un nombre de 2335 évènements ont été enregistrés dans l'ensemble de données et 2482 évènements supplémentaires provenant de sources inconnues sont également enregistrés dans l'ensemble de données.

Tableau IV-1: Caractéristiques du dataset BLUED

Caractéristiques des données	Multivariate time series
Types des données	Integer, Real, string
Niveau de détails	Toute la maison, les événements des équipements
Type de bâtiment	Résidentiel
Nombre de bâtiment	1
Adresse de bâtiment	Pennsylvanie, US
Date	2011
Durée	8 jours
Fréquence de prélèvement	12 khz

Le BLUED comprend un fichier ".txt" et 16 fichiers compressés bzip de 3.6 go chacun (location_001_datadictionary.txt et location001_dataset_001.bzip à 016). Le fichier du dictionnaire de données décrit les codes des appareils (par exemple 129 pour la télévision). Chacun des fichiers bzip contient des informations sur 12 heures par jour, notamment des données de consommation I-V (courant et tension), une liste d'événements et des listes d'événements pour les deux phases, les dates de début et de fin et un fichier Lisez-moi.

- Les attributs
- Date (aa / mm / jj)
- Heure (hh: mm: ss)
- Courant A (en ampères)
- Courant B (en ampères)
- Voltage A (en Volts)
- Puissance active (en watts)

IV.3.2 Extraction des évènements

Pour réduire la taille du dataset BLUED on était obligé d'extraire un sous-ensemble de données qui contient que les évènements et leurs étiquettes. Cette étape consiste à parcourir toute le dataset pour collecter les enregistrements (records) des événements dans Current et chaque enregistrement est sous forme d'une fenêtre d'une seconde dans laquelle il y a 12 000 samples ou l'évènement doit être dans le milieu de la fenêtre extraite.

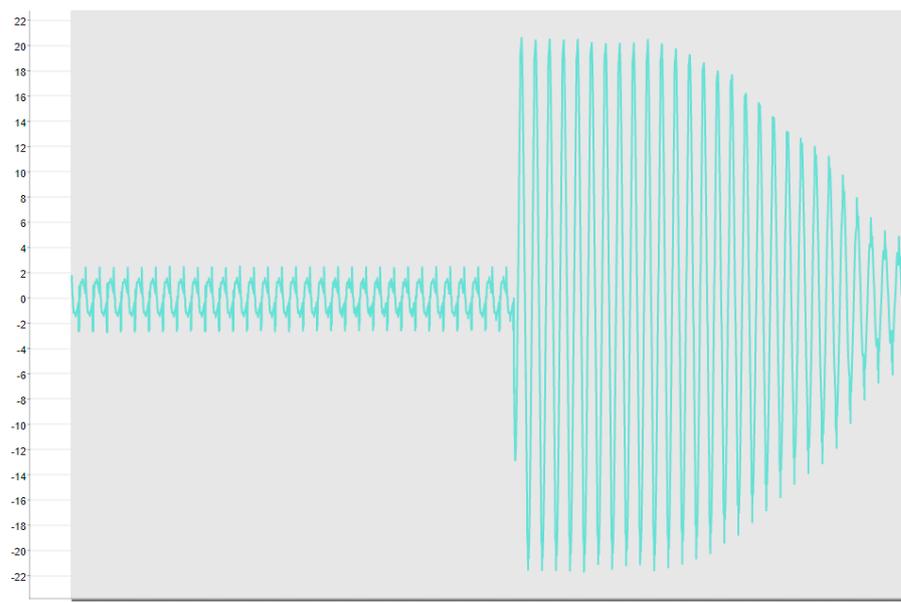


Figure IV-3: Un exemple d'une fenêtre extraite d'un évènement Current

IV.4 Prétraitement des données

Pour avoir une bonne modélisation des données et de bons résultats d'entraînement d'un modèle d'apprentissage automatique, les données doivent être prés-traitées pour éliminer les valeurs manquantes, erronées et les bruits. Pour cela on a appliqué quelques méthodes de traitement de signal qui sont non stationnaires (dont les caractéristiques spectrales varient au cours du temps).

IV.4.1 Fast Fourier Transform

A partir du signal temporel, il est possible d'effectuer une analyse spectrale qui permet de mettre en évidence les caractéristiques fréquentielles du signal. L'outil d'analyse nécessaire à ce type d'étude est la transformée de Fourier (dont l'application informatique se nomme FFT : Fast Fourier Transform). Sa complexité est de l'ordre de $\theta(n \log n)$ avec le nombre n de points, alors que la complexité de l'algorithme "naïf" est $\theta(n^2)$. Ainsi, pour $n = 1\ 024$, le temps de calcul de l'algorithme basé sur la transformée de Fourier peut être 100 fois plus court que celui de l'algorithme "naïf".

Pour établir cette étape on a utilisé le nœud FFT dans KNIME. Ce nœud effectue une transformation de Fourier rapide (FFT) vers l'avant sur chaque ligne de la table d'entrée (fenêtre). D'une manière générale, il extrait les fréquences d'un signal d'entrée. Le nombre de

colonnes de la table doit être une puissance de 2, c'est-à-dire 2, 4, 8, 16, 32, 64, ... Pour ce faire, donc on a réduit la taille de fenêtre pour qu'elle puisse avoir une taille avec une puissance de 2.

Fast Fourier Transform (FFT)

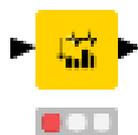


Figure IV-4: Le nœud Fast Fourier Transform (FFT)

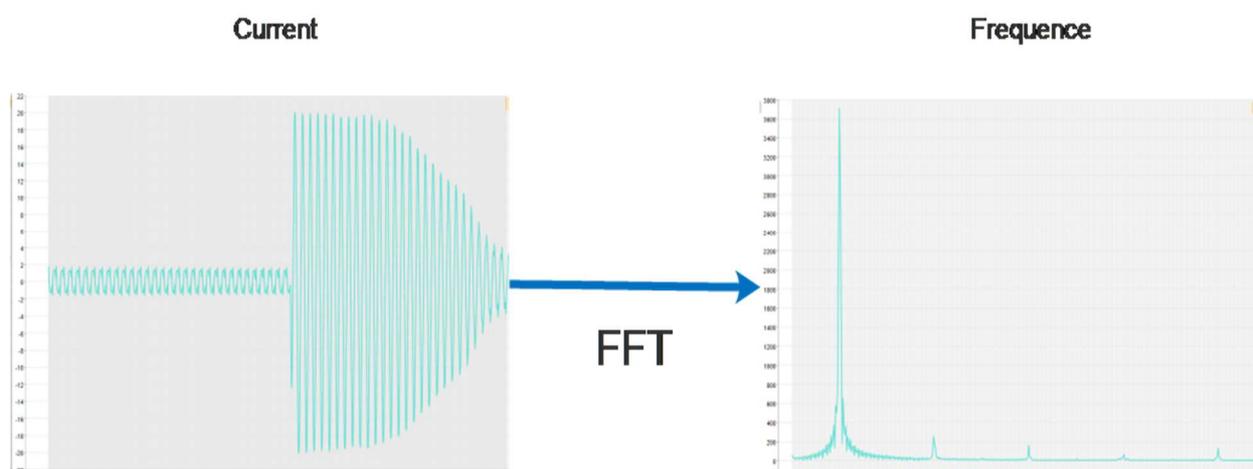


Figure IV-5: Une illustration d'un FFT

Il y a $\#$ (colonnes d'entrée) / 2 + 1 colonnes de sortie, appelées bacs de fréquences. Par exemple, si nous avons 256 colonnes d'entrée, nous obtiendrons 129 colonnes de sortie. Pour que ces groupes de fréquences soient correctement étiquetés, nous devons configurer le nœud avec la fréquence d'échantillonnage correcte.

IV.4.2 Normalisation des valeurs

Dans cette étape, on a utilisé une normalisation de Min-Max entre 0 et 1 qui consiste à faire une transformation linéaire de toutes les valeurs obtenues avec FFT de telle sorte que le minimum et le maximum dans chaque colonne soient présentés comme donnés.

Normalizer



Figure IV-6: Le noeud Normalizer

IV.5 Etapes D'apprentissage

Dans cette phase de projet, on a utilisé la méthode *DEC* dont l'entraînement de ce modèle d'apprentissage approfondi se compose en deux étapes. La première étape consiste à entraîner l'autoencodeur et la deuxième étape consiste à utiliser l'encodeur pour la classification automatique (clustering).

IV.5.1 Autoencodeur pré-entraîné

L'autoencodeur est un algorithme de compression de données dans lequel il y a deux parties principales : l'encodeur et le décodeur. Le travail de l'encodeur consiste à compresser les données d'entrée pour réduire les caractéristiques dimensionnelles. Par exemple, dans un échantillon d'un signal de 511 records au total, l'encodeur que nous avons élaboré, peut le compresser en un tableau avec seulement dix nombres à virgule flottante (image). La partie "décodeur", d'autre part, prend les caractéristiques compressées en entrée et reconstruit une image aussi près que possible de l'image d'origine.

L'autoencodeur est un algorithme d'apprentissage non supervisé dans la nature car pendant la phase d'entraînement, il ne prend que les images elles-mêmes et il n'a pas besoin d'étiquettes.

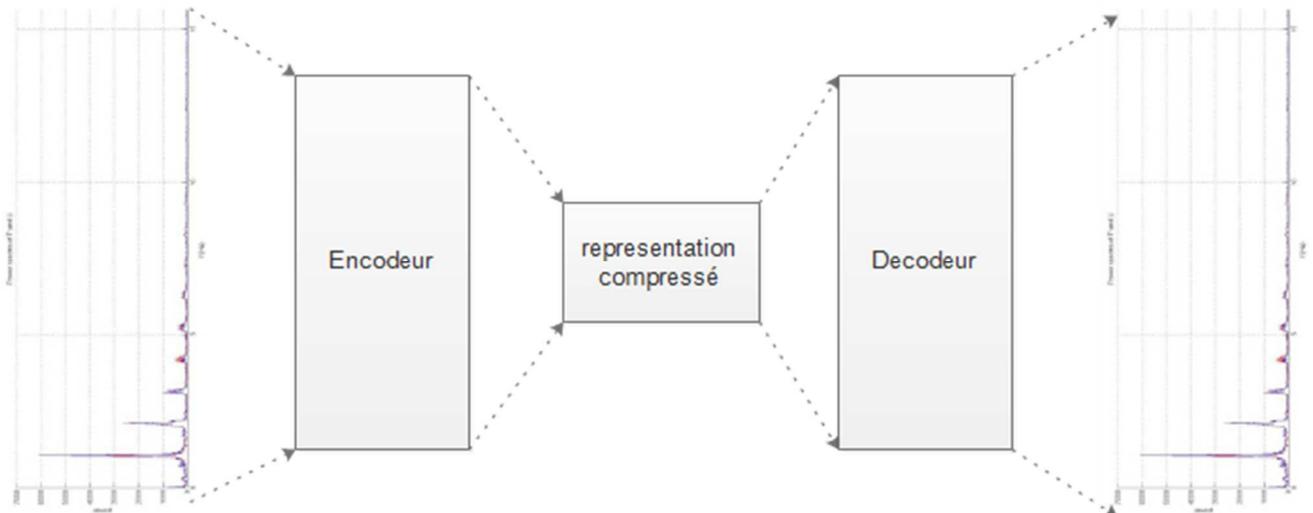


Figure IV-7 Une illustration d'un autoencodeur

L'Autoencodeur que nous construisons est un modèle symétrique entièrement connecté. Il est symétrique sur la façon dont un signal est compressé et décompressé par des manières opposées exactes. Le code source suivant illustre la schématisation de l'autoencodeur où x représente l'ensemble des fenêtres prétraitées.

```

def autoencoder(dims, act='relu', init='glorot_uniform'):
    n_stacks = len(dims) - 1
    # entrée
    input_img = Input(shape=(dims[0],), name='input')
    x = input_img
    # couches internes d'un encodeur
    for i in range(n_stacks-1):
        x = Dense(dims[i + 1], activation=act, kernel_initializer=init, name='encoder_%d' % i)(x)

    # couche de la representation compressée
    encoded = Dense(dims[-1], kernel_initializer=init, name='encoder_%d' % (n_stacks - 1))(x)
    x = encoded
    # couches internes d'un décodeur
    for i in range(n_stacks-1, 0, -1):
        x = Dense(dims[i], activation=act, kernel_initializer=init, name='decoder_%d' % i)(x)

    # sortie
    x = Dense(dims[0], kernel_initializer=init, name='decoder_0')(x)
    decoded = x
    return Model(inputs=input_img, outputs=decoded, name='AE'), Model(inputs=input_img, outputs=encoded, name='encoder')

dims = [x.shape[-1], 500, 500, 2000, 10]
init = VarianceScaling(scale=1. / 3., mode='fan_in',
                      distribution='uniform')
pretrain_optimizer = SGD(lr=1, momentum=0.9)
pretrain_epochs = 2000
batch_size = 256
save_dir = './results'

```

Figure IV-8: Le code source et le schéma de l'autoencodeur

Nous avons formé l'autoencodeur pour 2000 *époches* et enregistrer les poids du modèle pour les utiliser plus tard.

```

autoencoder.compile(optimizer=pretrain_optimizer, loss='mse')
autoencoder.fit(x, x, batch_size=batch_size, epochs=pretrain_epochs) #, callbacks=cb)
autoencoder.save_weights(save_dir + '/ae_weights.h5')

```

IV.5.2 Modèle de classification automatique

En entraînant l'autoencodeur, nous avons appris à sa partie encodeur à compresser chaque image en dix valeurs à virgule flottante. Nous pensons peut-être, puisque la dimensionnalité d'entrée est réduite à 10, K-Means devrait être capable de faire le clustering à partir d'ici. Pour cela, nous allons utiliser K-Means pour générer les centroïdes des clusters, qui sont les 10 centres des clusters dans l'espace des fonctionnalités 10-D. Mais nous allons également créer notre couche de clustering personnalisée pour convertir les entités en entrée en probabilité d'étiquette de cluster.

La probabilité est calculée par la distribution T de Student. Cette distribution est comme celle utilisée dans l'algorithme t-SNE. Elle mesure la similitude entre un point intégré et un centroïde. Ainsi, comme nous pouvons le deviner, la couche de clustering agit de manière similaire à l'approche K-means pour le clustering, et les poids de la couche représentent les

centroïdes des clusters qui peuvent être initialisés en entraînant selon l'approche K-means. Pour créer une couche personnalisée dans Keras, nous avons implémenté trois méthodes :

- **build(input_shape)** : où on a défini les poids de la couche. Dans notre cas, les 10 clusters dans l'espace d'entités 10-D, soit 10 x 10 variables de poids.
- **call(x)** : où réside la logique des couches, c'est là que se produit la correspondance entre les entités et la magie des étiquettes de clustering.
- **compute_output_shape(input_shape)** : on a spécifié ici la logique de transformation des formes d'entrée en sortie.

La figure IV-9 présente le code de couche de clustering personnalisé .

```
class ClusteringLayer(Layer):
    def __init__(self, n_clusters, weights=None, alpha=1.0, **kwargs):
        if 'input_shape' not in kwargs and 'input_dim' in kwargs:
            kwargs['input_shape'] = (kwargs.pop('input_dim'),)
        super(ClusteringLayer, self).__init__(**kwargs)
        self.n_clusters = n_clusters
        self.alpha = alpha
        self.initial_weights = weights
        self.input_spec = InputSpec(ndim=2)

    def build(self, input_shape):
        assert len(input_shape) == 2
        input_dim = input_shape[1]
        self.input_spec = InputSpec(dtype=K.floatx(), shape=(None, input_dim))
        self.clusters = self.add_weight(shape=(self.n_clusters, input_dim), initializer='glorot_uniform', name='clusters')
        if self.initial_weights is not None:
            self.set_weights(self.initial_weights)
            del self.initial_weights
        self.built = True

    def call(self, inputs, **kwargs):
        q = 1.0 / (1.0 + (K.sum(K.square(K.expand_dims(inputs, axis=1) - self.clusters), axis=2) / self.alpha))
        q **= (self.alpha + 1.0) / 2.0
        q = K.transpose(K.transpose(q) / K.sum(q, axis=1))
        return q

    def compute_output_shape(self, input_shape):
        assert input_shape and len(input_shape) == 2
        return input_shape[0], self.n_clusters

    def get_config(self):
        config = {'n_clusters': self.n_clusters}
        base_config = super(ClusteringLayer, self).get_config()
        return dict(list(base_config.items()) + list(config.items()))
clustering_layer = ClusteringLayer(n_clusters, name='clustering')(encoder.output)
model = Model(inputs=encoder.input, outputs=clustering_layer)
```

Figure IV-9: Code de la couche personnalisée

Ensuite, nous empilons une couche de clustering après l'encodeur préformé pour former le modèle de clustering. Pour la couche de clustering, nous initialisons ses poids. Les clusters étant centrés à l'aide de K-means et ils sont formés sur la base des vecteurs de caractéristiques de toutes les images.

L'étape suivante consiste à améliorer l'affectation de clustering et la représentation des fonctionnalités simultanément. À cette fin, nous définissons une distribution de probabilité cible

basée sur les centroïdes et nous minimisons sa divergence **KL** par rapport aux résultats du modèle de clustering.

Nous avons visé que la distribution cible ait les propriétés suivantes :

- Renforcer les prédictions, c'est-à-dire améliorer la pureté des clusters.
- Mettre davantage l'accent sur les points de données attribués avec une grande confiance.
- Empêcher les grands clusters de déformer l'espace des représentations compressées.

La distribution cible est calculée en élevant d'abord **q** (les vecteurs de représentations compressées) à la deuxième puissance, puis nous normalisons par fréquence et par cluster.

```
def target_distribution(q):  
    weight = q ** 2 / q.sum(0)  
    return (weight.T / weight.sum(1)).T
```

Il est nécessaire d'affiner d'une manière itérative les clusters en apprenant des affectations de confiance élevée à l'aide de la distribution cible auxiliaire. Après un nombre spécifique d'itérations, la distribution cible est mise à jour et le modèle de clustering est formé pour minimiser la perte de divergence **KL** entre la distribution cible et la sortie de clustering. La stratégie de formation peut être considérée comme une forme d'autoformation. Comme pour l'auto-formation, nous prenons un classificateur initial et un ensemble de données sans étiquette, puis nous étiquetons l'ensemble de données avec le classificateur pour nous entraîner sur ses prédictions de confiance élevée.

La fonction de perte, divergence **KL** ou divergence **Kullback – Leibler**, est une mesure de la différence de comportement entre deux distributions différentes. Nous voulons la minimiser afin que la distribution cible soit aussi proche que possible de la distribution de sortie de clustering.

```

for ite in range(int(maxiter)):
    if ite % update_interval == 0:
        q = model.predict(x, verbose=0)
        p = target_distribution(q) # update the auxiliary target distribution p

        # evaluate the clustering performance
        y_pred = q.argmax(1)
        if y is not None:
            acc = np.round(metrics.acc(y, y_pred), 5)
            nmi = np.round(metrics.nmi(y, y_pred), 5)
            ari = np.round(metrics.ari(y, y_pred), 5)
            loss = np.round(loss, 5)
            print('Iter %d: acc = %.5f, nmi = %.5f, ari = %.5f' % (ite, acc, nmi, ari), ' ; loss=', loss)

        # check stop criterion - model convergence
        delta_label = np.sum(y_pred != y_pred_last).astype(np.float32) / y_pred.shape[0]
        y_pred_last = np.copy(y_pred)
        if ite > 0 and delta_label < tol:
            print('delta_label ', delta_label, '< tol ', tol)
            print('Reached tolerance threshold. Stopping training.')
            break
        idx = index_array[index * batch_size: min((index+1) * batch_size, x.shape[0])]
        loss = model.train_on_batch(x=x[idx], y=p[idx])
        index = index + 1 if (index + 1) * batch_size <= x.shape[0] else 0

model.save_weights(save_dir + '/DEC_model_final.h5')

```

Figure IV-10: Extrait du code de la distribution cible

Dans l'extrait du code présenté dans la figure IV-10, la distribution cible est mise à jour toutes les 140 itérations de formation.

IV.6 Evaluation

La mesure indique qu'elle a atteint une précision de clustering de 56,8%, ce qui est bon compte tenu du fait que les entrées sont des signaux non étiquetés et le dataset est tout petit pour l'entraînement d'un modèle d'apprentissage approfondi. Examinons de plus près comment la précision est dérivée. Cette métrique prend une affectation prédite du cluster à partir d'un algorithme non supervisé et une affectation réelle, puis trouve la meilleure correspondance entre eux. Le meilleur mappage peut être calculé efficacement par *l'algorithme hongrois* qui est implémenté dans la bibliothèque d'apprentissage "*scikit*" en tant que `linear_assignment`. On peut remarquer dans la matrice de confusion que le nombre des instances utilisées pour l'entraînement et l'évaluation est tout petit.

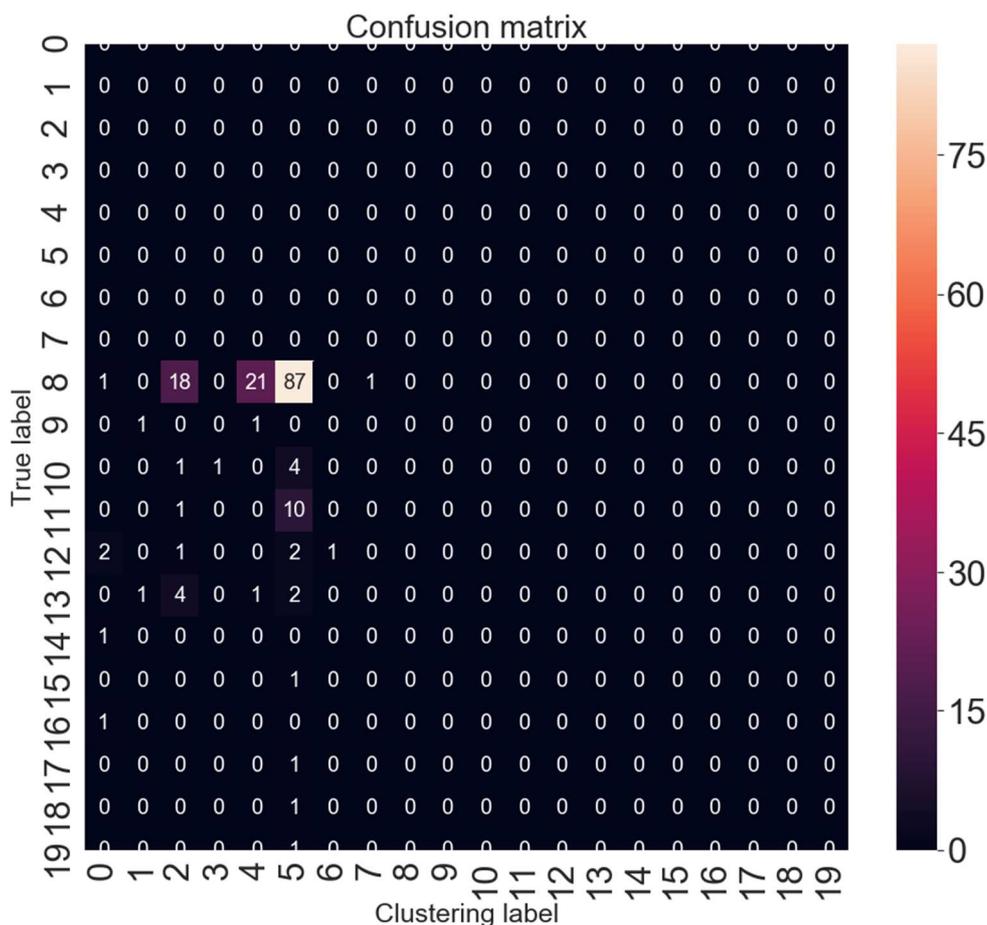


Figure IV-11: Résultats de la classification automatique

Note : La matrice de confusion est une matrice qui mesure la qualité d'un système de classification. Chaque ligne correspond à une classe réelle, chaque colonne correspond à une classe estimée. La cellule ligne L, colonne C contient le nombre d'éléments de la classe réelle L qui ont été estimés comme appartenant à la classe C₁. Un des intérêts de la matrice de confusion est qu'elle montre rapidement si un système de classification parvient à classifier correctement.

IV.7 Conclusion

Nous avons construit un modèle *DEC* avec la bibliothèque *Keras* pour effectuer une analyse de clustering avec des datasets 'non étiquetés'. L'autoencodeur prés-entraîné a joué un rôle très important dans la réduction dimensionnelle des données et l'initialisation des paramètres, puis la couche de clustering personnalisée a été formée en utilisant une distribution cible pour améliorer la précision.

Conclusion Générale

Conclusion générale

L'analyse de la consommation de l'énergie électrique et son impact sur notre entourage, nous a permis de conclure que la gestion optimale des charges des bâtiments, de façon active et intelligente, présente un fort enjeu, sachant que ces préoccupations majeures sont partagées par plusieurs acteurs à savoir les gestionnaires du réseau électrique, les fournisseurs, et les consommateurs.

En outre, il est plus que vrai de dire qu'aujourd'hui, le domaine de la domotique et le système de gestion d'énergie ont acquis une place fondamentale dans notre vie. Ces derniers visent à augmenter le niveau de confort tout en minimisant la consommation de l'énergie et protégeant l'environnement qui est le facteur le plus motivant pour l'évolution de ce domaine.

Sachant que, les différents acteurs tirent avantage de cette évolution, que ce soit le consommateur en matière de simplification de sa vie quotidienne et l'amélioration du confort, les gestionnaires des réseaux en matière d'amélioration de l'exploitation et de la fiabilité des réseaux ou les fournisseurs en matière d'adaptation des offres tarifaires.

Dans ce contexte, nous visons à améliorer le rendement de notre système et le doter par les techniques de machine learning pour apprendre sur les habitudes des habitants et proposer des solutions personnalisées à chaque habitant pour l'énergie et le confort. Dans le cadre de ce projet de fin d'études, nous avons proposé une solution qui consiste à détecter les états des équipements en traitant seulement la charge électrique agrégée. Pour ce faire, nous avons utilisé la méthode d'apprentissage profondi (DEC). Cette méthode de classification automatique se base sur un apprentissage profondi non supervisé. Nous avons pris comme environnement de test une maison surveillée dans laquelle il y a un ensemble d'équipements tels qu'une machine à laver, un lave-vaisselle, un chauffage et un climatiseur, etc. Par ailleurs, le but de la solution proposée est de créer des compteurs intelligents (soft sensors) qui permettent de connaître l'équipement qui est actif à un instant donné en fonction de la consommation de l'électricité.

Références Bibliographiques

Bibliographie

- [1] H. Kanchev, «Gestion des flux énergétiques dans un système hybride de sources d'énergie renouvelable,» Thèse de Doctorat, Ecole Centrale de Lille, France, 2015.
- [2] "Smartgrids-cre.fr : site d'information sur les Smart Grids", [En ligne]. Available: <http://www.smartgrids-cre.fr/index.php>. [Accès le 15 Avril 2020].
- [3] M. M. Begovic, *Electrical Transmission Systems and Smart Grids*, New York: Springer-Verlag New York, 2013.
- [4] A. Mercier, «Pilotage de la production décentralisée et des charges non conventionnelles dans le contexte Smart Grid,» Thèse de Doctorat, Université Grenoble Alpes, France, 2015.
- [5] "Réseaux électriques (Statistiques de l'électricité en France)", 24 Aout 2018. [En ligne]. Available: https://www.rte-france.com/sites/default/files/presentation_des_seef_2015.pdf.
- [6] M. Drouineau, «Introduction aux smart grids : réponse aux enjeux du secteur électrique,» Rapport, Département d'économie, Fonctionnement et Études des Systèmes Énergétiques, Paris, France, 2012.
- [7] M. Walid, «Réalisation d'un système de contrôle de l'énergie et de la sécurité des bâtiments,» Thèse de Doctorat, Université Abderahman Mira, Algeria, 2017.
- [8] François-XavierJeuland, "La maison communicante", Eyrolles, 2005.
- [9] N. H. NGUYEN, "Développement de méthodes intelligentes pour la gestion énergétique des bâtiments, utilisant des capteurs sans fil", Thèse de Doctorat, Université de Grenoble, 2009.
- [10] A. S.-K. GmbH, "Smart Home and Intelligent Building ControlEnergy Efficiency in Buildings with ABB i-bus® KNX", May 10 2018. [En ligne]. Available: https://library.e.abb.com/public/ecb0f775769e44ca9ae663e38687be10/2CDC500060M0201_HB_EnergyEfficiency_EN.pdf.
- [11] X.-C. Le, "Improving performance of non-intrusive load monitoring with low-cost sensor networks", Thèse de Doctorat, Université de Rennes, 2017.
- [12] M. Taffar, «Initiation à l'apprentissage automatique,» Support de Cours pour étudiants en Master en Intelligence Artificielle, Université de Jijel, 2018.
- [13] L. M. Y. A. Cornuéjols, «Apprentissage Artificiel, Concepts et algorithmes,» Eyrolles; Édition : 2 (3 juin 2010), 2002.
- [14] P. Vincent, «Modèles à noyaux à structure locale,» Thèse de Doctorat, Département d'informatique et de recherche opérationnelle, Université de Montréal, Canada, 2003.

- [15] B. Azeddine, «Apprentissage Incrémental,» Support de cours, Université Hadj Lakhdar, Batna, Algeria, 2013.
- [16] T. Z. Souad, «Contribution a l'apprentissage automatique symbolique par automates d'arbre et mesure de selection,» université Oran, 2014.
- [17] J. MacQueen, «Some methods for classification and analysis of multivariate observations,» *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, pp. 281-297, 1967.
- [18] S. C. Johnson, "Hierarchical Clustering Schemes", *Psychometrika*, vol. 32, pp. 241-254, 1967.
- [19] G. N. L. a. W. T. Williams, "A General Theory of Classificatory Sorting Strategies. I. Hierarchical Systems", *The Computer Journal*, vol. 9, pp. 373-380, 1967.
- [20] G. Christian, "Apprentissage profond", Université de LAVAL, Canada, 2017.
- [21] L. Stanislas, «Exploration des réseaux de neurones à base d'autoencodeur dans le cadre de la Modélisation des données textuelles,» Thèse de Doctorat, Université de Sherbrooke, Canada, 2016.
- [22] H. Tolgui, "Deep learning par reconnaissances du visage", Mémoire de Master, Université du Biskra, Algérie, 2018.
- [23] L. P. A. Manuel, "Le deep learning pas à pas", PARTIE I : Concepts, Des labos de R&D à la vie quotidienne. SQLI Digital experience", [En ligne]. Available: <https://www.technologies-ebusiness.com/enjeux-et-tendances/le-deep-learning-pas-a-pas>. [Accès le 15 April 2020].
- [24] Y. M. Djaloul, "Deep Learning pour la classification des images", Mémoire de Master, Département d'Informatique, Université de Tlemcen, 2017.
- [25] B. Valentin, «Approches Deep Learning Appliquées aux signaux audio", Mémoire de Master, Institut de Recherche en Informatique de Toulouse, France, 2015.
- [26] G. E. O. S. & T. Y. W. Hinton, "A fast learning algorithm for deep belief nets", *Neural Computation*, vol. 18, n° 17, pp. 1-16, 2006.
- [27] A. Labiad, "Sélection des mots clés basée sur la classification et l'extraction des règles d'association", Mémoire de Master, Université du Québec à Trois-Rivières, 2017.
- [28] "Clustering Autoencoders: Comparing DEC and DCN"? [En ligne]. Available: <https://luckytoilet.wordpress.com/2019/07/07/clustering-autoencoders-comparing-dec-and-dcn/>. [Accès le 7 July 2019].
- [29] "L'Internet des Objets pour la santé", Alcatel.Lucent.
- [30] «datatracker.ietf.org,» 2017. [En ligne]. Available: <https://datatracker.ietf.org/wg/roll/charter/>.
- [31] L. M. Y. A. Cornuéjols, "Apprentissage Artificiel, Concepts et algorithmes", 2002.

- [32] L. Stanislas, «Exploration des réseaux de neurones à base d'autoencodeur dans le cadre de la Modélisation des données textuelles,» université de Sherbrooke, 2016.

Résumé

Les solutions de surveillance de l'énergie dans les maisons intelligentes proposées ces dernières années convergent vers des méthodes d'apprentissage non supervisées qui peuvent être adaptables et scalables.

Dans le cadre de ce travail, nous avons proposé une solution qui permet de détecter les états des équipements en traitant seulement la charge électrique agrégée. Pour cela, nous avons utilisé la méthode d'apprentissage approfondi (DEC : Deep Clustering). Cette méthode de classification automatique se base sur un apprentissage approfondi non supervisé. Nous avons pris comme environnement de validation une maison surveillée dans laquelle il y a un ensemble d'équipements (une machine à laver, une lave-vaisselle, un chauffage et un climatiseur, etc). Par ailleurs, le but de la solution proposée est de créer des compteurs intelligents qui permettent de connaître l'équipement qui est actif à un instant donné en fonction de la consommation de l'électricité. Cette solution se base sur un nouveau mécanisme de clustering des images qui permet la réduction de la dimension du problème traité en utilisant les "AutoEncoders" et la plateforme KNIME.

Mots Clés : Deep Clustering, KNIME, AutoEncoders, Apprentissage non supervisé, Réseaux de capteurs.

Abstract

Smart home energy monitoring solutions provided in recent years converge on unsupervised learning methods that can be adaptable and scalable.

In this work, we proposed a solution which makes it possible to detect the states of the equipment by treating only the aggregated electrical charge. For this, we used the deep learning method (DEC: Deep Clustering). This automatic classification method is based on in-depth unsupervised learning. We used as a validation environment a monitored house in which there is a set of equipments (a washing machine, a dishwasher, a heater and an air conditioner, etc.). In addition, the goal of the proposed solution is to create smart meters which allow to know the equipment which is active at a given time according to the consumption of electricity. This solution is based on a new image clustering mechanism which allows the reduction of the dimension of the problem treated by using "AutoEncoders" and the KNIME platform.

Keywords: Deep Clustering, KNIME, AutoEncoders, Unsupervised learning, Sensor networks.

ملخص

تتقارب حلول مراقبة الطاقة في المنازل الذكية المقترحة في السنوات الأخيرة على طرق التعلم غير الخاضعة للإشراف والتي يمكن أن تكون قابلة للتكيف وقابلة للتطوير.

في إطار هذا العمل ، اقترحنا حلاً يجعل من الممكن اكتشاف حالات المعدات من خلال معالجة الشحنة الكهربائية المجمعة فقط. لذلك ، استخدمنا طريقة التعلم العميق (DEC: Deep Clustering). تعتمد طريقة التصنيف التلقائي المستعملة على التعلم المتعمق غير الخاضع للرقابة. استخدمنا كبنية للتحقق من الصحة منزلاً مراقباً توجد فيه مجموعة من المعدات (غسالة ، غسالة صحون ، سخان ومكيف هواء ، إلخ). بالإضافة إلى ذلك ، فإن الهدف من الحل المقترح هو إنشاء عدادات ذكية تسمح بمعرفة المعدات النشطة في وقت معين وفقاً لاستهلاك الكهرباء. يعتمد هذا الحل على آلية جديدة لتجميع الصور تسمح بتقليل أبعاد المشكلة التي يتم معالجتها باستخدام "AutoEncoders" ومنصة KNIME.

الكلمات المفتاحية: التجمعات العميقة، KNIME، الترميز التلقائي، التعلم بدون إشراف، شبكات أجهزة الاستشعار.