SUP'COM
Ecole Supérieure des Communications de Tunis

جامعة قرطاج
Université de Carthage

Rapport de stage d'ingénieur

Prédiction de la consommation d'électricité des usines de fabrication

Organisme d'accueil :

Sagemcom

Réalisé par :

Nouha BEN HAMADA

Encadré par :

Mr. Amine MEZGHANI

Année universitaire : 2023/2024

Signatures

Remerciement

Ce rapport est le fruit d'un excellent stage chez Sagemcom SST, grâce à l'interaction que j'ai eue avec mon encadrant, collègues et collaborateurs.

Je tiens à exprimer mes remerciements particuliers à mon encadrant, M. Amine Mezghani, qui m'a guidée tout au long de ce merveilleux projet et m'a aidée dans la recherche et l'élaboration des différentes approches.

Enfin, je remercie l'ensemble des employés de Sagemcom SST pour les conseils qu'ils ont pu me prodiguer tout au long de la durée du stage.

Résumé

Ce stage s'inscrit dans le cadre de notre cursus académique du cycle d'ingénieur. L'objectif de ce travail est de mettre en pratique la théorie qu'on étudie tout au long de deux années à SupCom.

Le travail demandé par l'organisme d'accueil SAGEMCOM est de développer un modèle de prédiction de la consommation d'électricité des usines de fabrication. À l'aide des données, collectées à des intervalles d'une minute sur une période de sept mois, comprenant la consommation totale d'énergie des usines, les types de fabrication, les dates de participation à des programmes de réponse à la demande (DR), ainsi que les capacités de réduction et de réponse des usines.

Mots clés : Python , Prédiction , Machine Learning, Deep Learning, Clustering.

Table des matières

Liste des figures

Liste des tableaux

Introduction Générale

La transition vers des sources d'énergie renouvelables est essentielle pour atteindre les objectifs de neutralité carbone et lutter contre le changement climatique. Cependant, cette transition pose des défis pour la stabilité des réseaux électriques en raison des variations imprévisibles de l'offre et de la demande. Pour maintenir cet équilibre, la prédiction précise de la consommation d'énergie, notamment dans les secteurs industriels, est cruciale.

Ce projet vise à développer des modèles de prédiction de la consommation d'électricité des usines de fabrication. En utilisant des données collectées à des intervalles d'une minute sur sept mois, incluant la consommation totale d'énergie, les types de fabrication, les dates de participation à des programmes de réponse à la demande (DR), ainsi que les capacités de réduction et de réponse des usines.

J'ai effectué ce projet au sein de SAGEMCOM SST. Ce rapport vise dans un premier temps à présenter l'entreprise d'accueil et l'idée générale du projet, par la suite nous allons passer à une étude théorique dans laquelle on présentera le contexte du projet et les grandes notions du projet. Nous décrirons ensuite l'implémentation et les résultats de ce stage et on finira par conclure.

Chapitre 1 : Le contexte du stage

1.1 Introduction

Tout au long de ce chapitre, on va commencer par présenter l'organisme d'accueil, décrire le groupe SAGEMCOM, parler de ses activités et son organisation. Par la suite, on va présenter le projet en décrivant la problématique et la solution proposée.

1.2 Présentation de l'entreprise d'accueil

1.2.1 Description de l'entreprise

Sagemcom [1] est un groupe industriel français, leader mondial des produits et solutions communicantes à destination des marchés du Broadband, des solutions Audio et Vidéos, et de l'énergie (électricité, gaz, et eau). Sagemcom conçoit, fabrique et expédie ses produits partout le monde, grâce à des usines en propre et à des partenaires industriels présents sur tous les continents. L'effectif de 6 500 personnes est réparti dans plus de 50 pays. 30% du capital de Sagemcom est détenu par ses collaborateurs.

1.2.2 Activités de l'entreprise

Grâce à sa culture commune fondée sur l'innovation, les progrès technologiques, la valeur ajoutée élevée et le temps nécessaire au marché, SST développe et produit une gamme complète de produits qui répondent aux marchés en forte croissance, en particulier dans trois grands marchés :

- Broadband : SAGEMCOM Broadband est l'un des principaux fabricants européens de SetTop Boxes et de portes résidentielles. Il propose des produits présentant les toutes dernières avancées technologiques, y compris les Set-Top Boxes qui sont compatibles avec tous les écosystèmes de la télévision, des passerelles résidentielles offrant un accès gigabit de bout en bout, et des Boxes combinant l'accès à large bande et à la télévision dans un seul produit.
- Smart City : SAGEMCOM possède plus de 50 ans d'expérience dans la conception et le déploiement de projets d'infrastructures de télécommunications (infrastructures de réseau et gestion de l'information), et plus de 20 ans d'expérience dans la mesure de l'énergie et de la mesure intelligente (eau, gaz et électricité), en déployant des solutions sûres et durables.
- Internet des Objets (IOT) : SAGEMCOM propose une offre intégrée de bout en bout adaptée à l'Internet industriel des objets, basée sur la norme ouverte LoRaWAN. Cette offre couvre les modules radiofréquences utilisés pour connecter les éléments, jusqu'à l'ensemble du réseau d'infrastructure permettant à la fois la réception des données transmises par les capteurs, et le contrôle de ces éléments en émettant des commandes du réseau.

1.2.3 Organisation de l'entreprise

L'organisation de la SST consiste en une orientation opérationnelle, un service des ressources humaines, des services généraux, un service de qualité et les trois pôles suivants :

- Pôle d'activités Décodeurs et Télévisions : Ce pôle comprend de nombreuses activités telles que la conception, le développement et la validation de logiciels embarqués STB, et l'intégration de briques de contrôle d'accès sur le décodeur.
- Terminaux résidentiels Pôle d'activité : Ce pôle comprend de nombreuses activités telles que la conception, le développement et la validation de logiciels embarqués pour des passerelles résidentielles et professionnelles, offrant des services VoIP et la télévision, utilisant les technologies xDSL et FTTH.
- Pôle Énergie et télécommunications : Ce pôle est ouvert récemment, il assure la conception, le développement et la validation de logiciels embarqués de compteurs d'électricité numérique, et il assure aussi l'innovation pour l'internet des objets basée sur la norme ouverte LoRaWAN.

1.3 Présentation du projet

1.3.1 Présentation de la problématique

Pour atteindre la neutralité carbone, les industries doivent intégrer de manière significative les énergies renouvelables dans leurs opérations. Cependant, cette intégration pose des défis pour l'équilibre entre la demande et l'offre d'énergie, en raison de la nature intermittente des sources renouvelables comme le solaire et l'éolien. Dans les usines de fabrication, de grands consommateurs d'énergie, cette variabilité complique la prévision des besoins énergétiques, rendant difficile la gestion efficace de la consommation pour soutenir la stabilité du réseau électrique.

La problématique centrale du projet est donc de développer un modèle de prédiction de la consommation d'énergie pour les usines de fabrication. Cela permettra d'améliorer l'équilibre entre l'offre et la demande énergétique, contribuant ainsi aux objectifs de neutralité carbone tout en assurant la fiabilité du système électrique.

1.3.2 Présentation de la solution proposée

La solution proposée vise à développer un modèle de prédiction de la consommation d'énergie des usines de fabrication en utilisant des techniques de machine learning. Ce modèle analysera des données collectées à des intervalles d'une minute sur une période de sept mois, incluant la consommation totale d'énergie des usines, les types de fabrication, les dates de participation aux programmes de réponse à la demande [2] (DR), ainsi que les capacités de réduction et de réponse des usines. Ces données proviennent de la base de données de l'Institut coréen d'évaluation et de planification des technologies énergétiques, offrant un ensemble riche et pertinent pour la modélisation.

1.4 Conclusion

Ce chapitre a été consacré à décrire l'organisme d'accueil et le projet comme une étape d'introduction qui peut aider à mieux comprendre le milieu du travail.

Chapitre 2 : Approche du projet

2.1 Introduction

Dans ce chapitre, on va faire une étude théorique des technologies utilisées dans ce projet, tels que le clustering, les modèles de régression et d'apprentissage profond. Par la suite, on va décrire les étapes de ce projet. Et en fin, on va présenter les métriques d'évaluation.

2.2 Etude théorique

2.2.1 Les techniques de clustering temporel

Le clustering à caractère temporel regroupe des séries temporelles en fonction de leurs similarités dans le temps. Contrairement au clustering classique, il prend en compte la dimension temporelle des données, capturant ainsi des motifs récurrents et des tendances. Cette méthode est idéale pour analyser des données comme les variations de la consommation d'énergie ou les tendances de marché, où les comportements évoluent au fil du temps.

TimeSeriesKMeans [3] est une variante du k-means spécifiquement conçue pour les séries temporelles. Il utilise des mesures de distance adaptées, comme la Dynamic Time Warping (DTW), pour regrouper des séries présentant des motifs similaires malgré les décalages temporels. Cet algorithme ajuste les centroides des clusters en fonction des séries temporelles assignées, facilitant ainsi l'analyse des comportements chronologiques complexes.

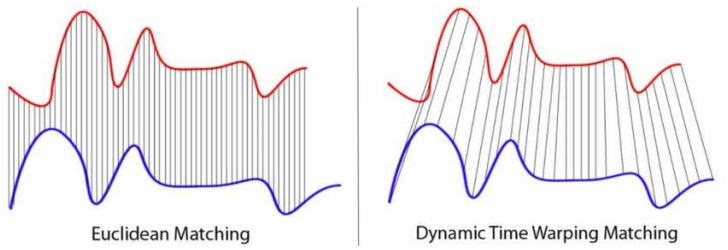

Figure 2.1 : La différence entre la distance euclidienne et le DTW

2.2.2 Les méthodes de régression

- **Random Forest Regressor** [4] est un algorithme d'apprentissage automatique supervisé utilisé pour les tâches de régression. Il fonctionne en combinant plusieurs arbres de décision indépendants, chacun formé sur des sous-ensembles aléatoires des données et des variables. En agrégeant les prédictions de ces différents arbres (en moyenne), le modèle réduit le risque de surapprentissage et

améliore la précision globale. Grâce à cette approche en ensemble (ensemble learning), le Random Forest Regressor est capable de modéliser des relations complexes entre les variables et est particulièrement robuste face aux données bruitées ou incomplètes.

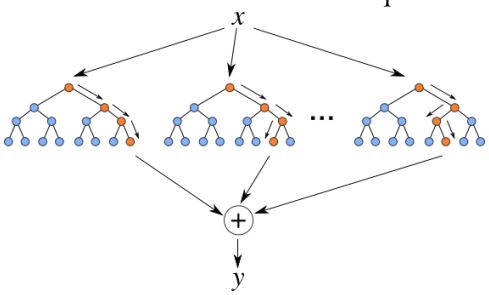

Figure 2.2 : Random Forest Regressor

- **Gradient Boosting Regressor** [5], à la différence du Random Forest Regressor, construit les arbres de manière séquentielle, où chaque nouvel arbre corrige les erreurs commises par les précédents. L'idée principale est d'optimiser les performances en minimisant progressivement l'erreur résiduelle à chaque itération, en ajustant les prédictions de chaque nouvel arbre aux écarts des prédictions précédentes. Ce processus de "boosting" permet d'atteindre une plus grande précision, bien que le modèle puisse être plus sensible au surapprentissage que les forêts aléatoires. Cependant, lorsqu'il est bien régularisé, le Gradient Boosting Regressor offre une performance puissante pour des tâches complexes de régression.

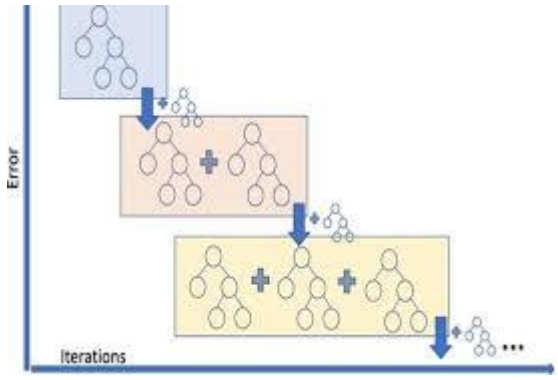

Figure 2.3 : Gradient Boosting Regressor

- **XGBoost** (Extreme Gradient Boosting) [6] est une implémentation optimisée de l'algorithme de Gradient Boosting, conçue pour offrir de meilleures performances et une plus grande efficacité. Par rapport au Gradient Boosting classique, XGBoost intègre plusieurs améliorations, telles que la régularisation explicite pour prévenir le surapprentissage, l'optimisation parallèle pour accélérer les calculs, et des techniques avancées de gestion des données manquantes. Grâce à ces optimisations, XGBoost est souvent plus rapide et performant, notamment sur des jeux de données volumineux et complexes, tout en conservant la capacité à minimiser les erreurs de manière progressive.

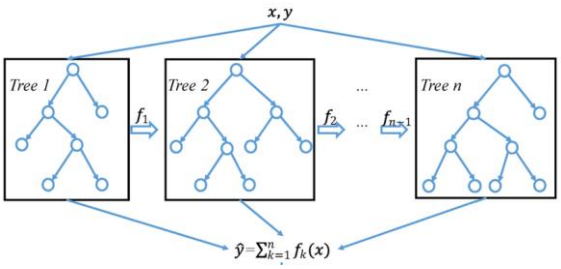

Figure 2.4 : Extreme Gradient Boosting

2.2.3 Les modèles d'apprentissage profond

- Les **RNN** [7] (réseaux de neurones récurrents) sont des réseaux de neurones conçus pour traiter des données séquentielles, comme les séries temporelles ou le texte. Contrairement aux réseaux classiques, les RNN possèdent des connexions récurrentes qui leur permettent de "se souvenir" des informations issues des étapes précédentes, créant ainsi une forme de mémoire interne. Cette capacité à conserver le contexte sur plusieurs étapes rend les RNN efficaces pour les tâches où l'ordre des données est crucial, comme la traduction automatique, la reconnaissance vocale, ou la prévision de séries temporelles. Cependant, ils souffrent souvent de problèmes comme l'oubli à long terme, ce que des variantes comme les LSTM cherchent à résoudre.

- **LSTM (Long Short-Term Memory)** [8] est un type de réseau de neurones récurrent (RNN) spécialement conçu pour traiter et prédire des données séquentielles avec des dépendances à long terme. Contrairement aux RNN classiques, qui peuvent avoir du mal à se souvenir des informations sur de longues séquences, l'architecture LSTM utilise des cellules de mémoire et des portes (d'entrée, de sortie et d'oubli) pour mieux contrôler le flux d'informations. Cela permet au modèle de conserver des informations pertinentes sur des périodes prolongées et d'ignorer celles qui ne sont plus utiles. Les LSTM sont particulièrement efficaces pour des tâches comme la prévision de séries temporelles, la traduction automatique et la reconnaissance vocale.

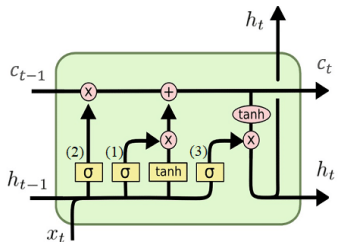

Figure 2.5 : Une entité LSTM

- Le **GRU** (Gated Recurrent Unit) [9] est une variante simplifiée des réseaux LSTM, conçue pour traiter des données séquentielles tout en étant plus rapide et plus facile à entraîner. Comme les LSTM, les GRU utilisent des mécanismes de portes pour contrôler le flux d'informations à travers le réseau, mais avec une structure plus simple, fusionnant les portes d'entrée et d'oubli en une seule. Cette simplification permet aux GRU de gérer efficacement les dépendances à long terme tout en nécessitant moins de ressources computationnelles. Les GRU sont souvent utilisés dans des tâches comme la prédiction de séries temporelles et le traitement du langage naturel, offrant des performances similaires à celles des LSTM avec une complexité réduite.

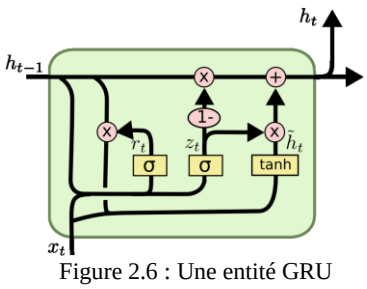

Figure 2.6 : Une entité GRU

2.3 Etapes du projet

Ce sont les étapes nécessaires pour la prédiction de la consommation d'énergie électrique des usines de fabrication :

1. **Collecte des données** :
   La première étape consiste à rassembler les données pertinentes pour construire un modèle prédictif robuste. Les données peuvent provenir de diverses sources, et dans ce projet, elles sont issues de la base de données de l'Institut coréen d'évaluation et de planification des technologies énergétiques. Ces données comprennent les informations sur la consommation énergétique des usines, leur participation aux programmes de réponse à la demande (DR), ainsi que leurs capacités de réduction et de réponse énergétique.

2. **Prétraitement des données** :
   Le prétraitement des données est crucial pour garantir la qualité et la précision des modèles d'apprentissage automatique. Il inclut le nettoyage des données brutes en traitant les valeurs manquantes, les données bruyantes et les incohérences. Ce processus vise à préparer un ensemble de données cohérent et exploitable pour l'entraînement des modèles.

3. **Clustering** :
   Une fois les données nettoyées, il est pertinent de regrouper les usines selon leurs profils de consommation énergétique. Le clustering permet de diviser les usines en groupes ou clusters ayant des comportements similaires. Cela facilite la prédiction de la consommation d'énergie pour chaque groupe, car les usines au sein d'un même cluster partagent des caractéristiques de consommation analogues.

4. **Recherche de modèle** :
   L'objectif de cette étape est de sélectionner et de former le modèle le plus performant possible à partir des données prétraitées. Pour cela, il est nécessaire de bien comprendre la nature des données et de déterminer le type de problème à résoudre. Il est également important de prendre en compte les contraintes, telles que la capacité de stockage des données et les performances souhaitées en termes de précision et de vitesse d'apprentissage.

5. **Entraînement et test** :
   À cette étape, le jeu de données est divisé en deux ensembles : l'un pour l'entraînement du modèle et l'autre pour les tests. L'objectif est de construire un modèle prédictif et d'ajuster ses paramètres afin de maximiser sa performance. La validation croisée peut être utilisée pour évaluer la généralisation du modèle sur des données non vues.

6. **Évaluation** :
   Enfin, l'évaluation permet de déterminer dans quelle mesure le modèle sélectionné est capable de représenter fidèlement les données et de prédire la consommation d'énergie future. Des critères de performance, tels que l'erreur quadratique moyenne (MSE) ou le coefficient de détermination (R^2), peuvent être utilisés pour comparer différents modèles et sélectionner celui offrant les meilleures performances.

2.4 Métriques d'évaluation

Dans ce projet, la prédiction de la consommation d'énergie électrique des usines de fabrication est un **problème de régression**, où l'objectif est de prédire une valeur numérique continue. Par conséquent, le choix du modèle le plus approprié dépendra de la capacité à minimiser l'erreur entre les valeurs réelles et prédites. Pour évaluer cette performance, nous utiliserons les métriques suivantes :

- **Erreur Quadratique Moyenne (MSE)** :

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Le MSE mesure la moyenne des carrés des écarts entre les valeurs réelles et les valeurs prédites, en amplifiant les grandes erreurs.

- **Erreur Quadratique Moyenne Racine (RMSE)** :

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Le RMSE est la racine carrée du MSE, exprimant les erreurs dans les mêmes unités que les données, ce qui facilite leur interprétation.

- **Coefficient de Détermination (R^2)** :

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Le R^2 indique la proportion de la variance des données expliquée par le modèle. Une valeur proche de 1 signifie que le modèle prédit bien les données.

2.5 Conclusion

Dans ce chapitre, nous avons commencé par clarifier le contexte théorique du projet. Puis nous avons expliqué les étapes de déroulement du projet et nous avons mis en place les métriques d'évaluation.

Chapitre 3 : Implémentation et résultats

3.1 Introduction

Tout au long ce chapitre, on va commencer par présenter les technologies et les bibliothèques utilisées pour réaliser ce projet, par la suite visualiser et préparer les données, le Clustering. Et en fin, on l'entrainement et l'évaluation des modèles.

3.2 Outils utilisés

Pour la réalisation de ce projet, nous avons utilisé une combinaison d'outils et de bibliothèques Python afin de développer, tester et évaluer les modèles de prédiction. Les principaux outils utilisés sont les suivants :

- **Python** : Langage de programmation principal pour le développement des modèles et le traitement des données.
- **Jupyter Notebook** : Environnement interactif pour le codage et la documentation des analyses.
- **TensorFlow** : Bibliothèque pour le développement et l'entraînement des modèles d'apprentissage profond.
- **Pandas** : Bibliothèque pour la manipulation et l'analyse des données.
- **scikit-learn** : Bibliothèque pour les algorithmes de machine learning et les outils de prétraitement des données.
- **NumPy** : Bibliothèque pour le calcul scientifique et la manipulation de tableaux de données.
- **Matplotlib** : Bibliothèque pour la visualisation des données et des résultats.

Ces outils ont permis de couvrir l'ensemble du processus de modélisation, du prétraitement des données à l'évaluation des performances des modèles.

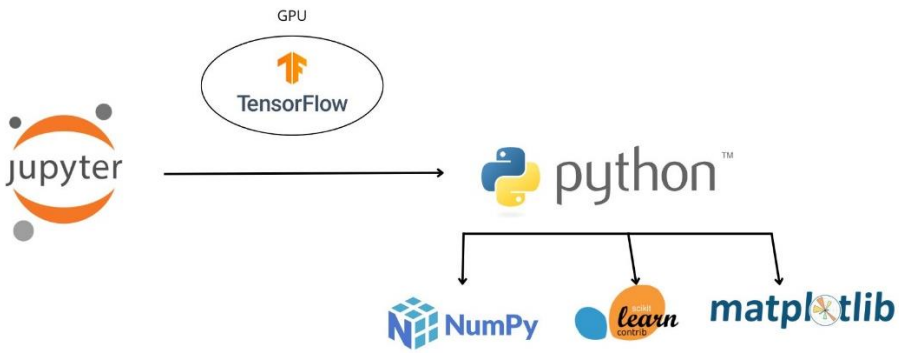

Figure 3.1 : Les technologies utilisées

3.3 Visualisation des données

Les données utilisées dans ce projet comprennent dix bases de données distinctes, chacune enregistrant la consommation d'énergie électrique en kilowatts (kW) à chaque minute pendant une période de dix mois, allant du 1er mars 2019 au 30 septembre 2019.

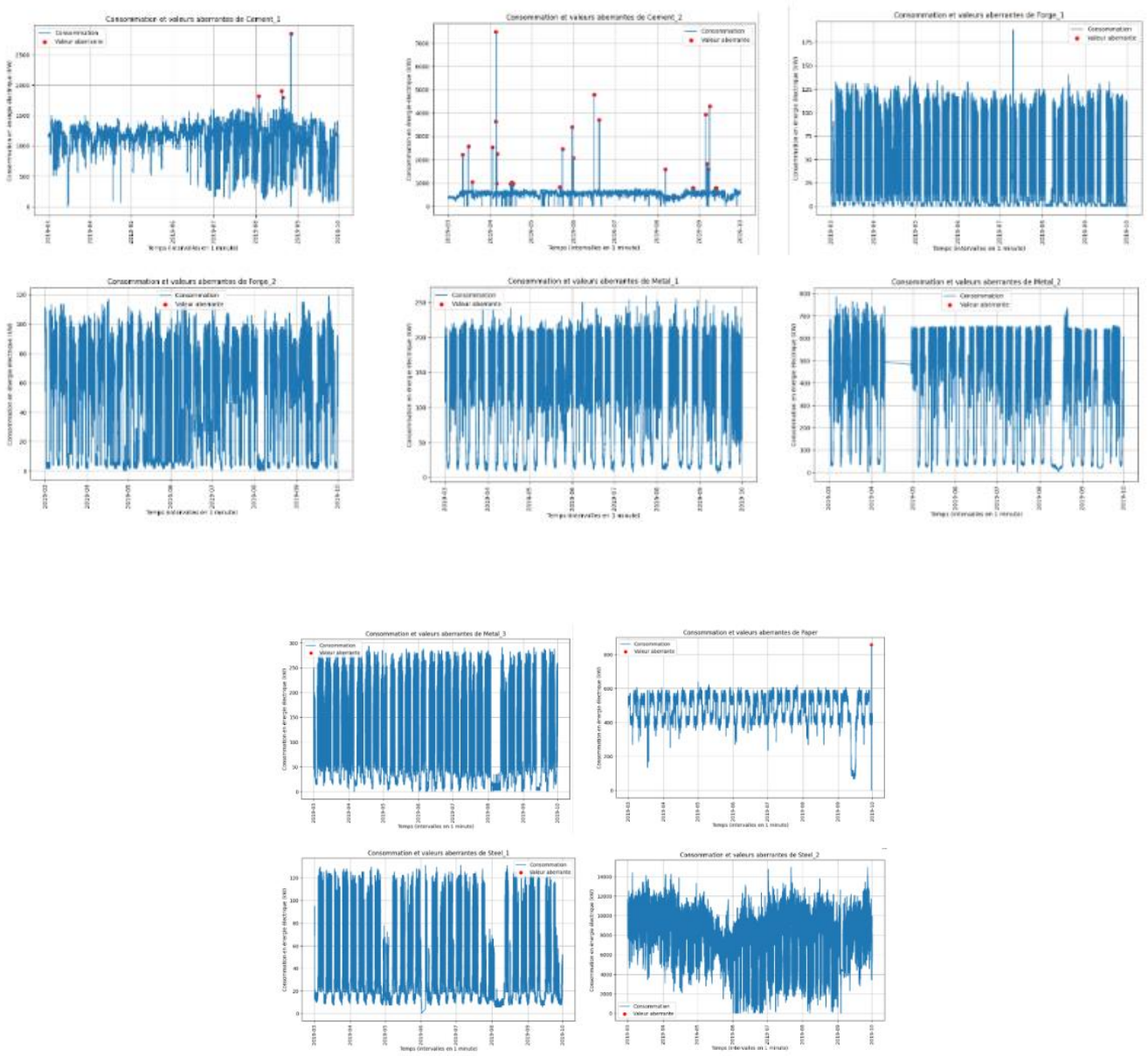

Figure 3.2 : Visualisation de la consommation énergétique des usines au cours du temps

Ces usines présentent un cycle de consommation régulier chaque jour de la semaine, avec des horaires presque fixes durant lesquels elles augmentent ou diminuent leur consommation d'énergie. Ce comportement est illustré dans la figure ci-dessous.

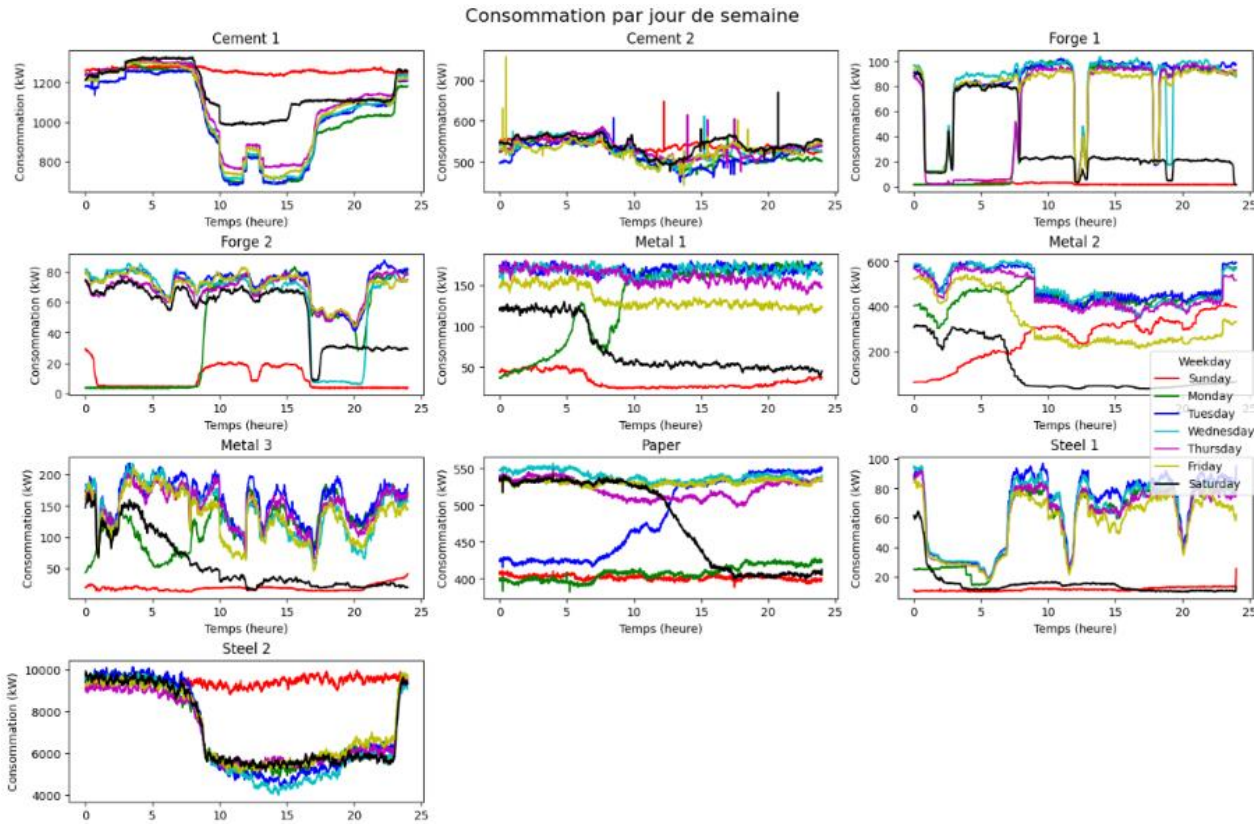

Figure 3.3 : Visualisation de la consommation par jour de semaine des usines de fabrication

En outre, une base de données complémentaire fournit des informations sur la participation de ces dix usines à un programme de réponse à la demande, où les usines ont réduit leur consommation énergétique pour aider à équilibrer la demande du système électrique.

Usine	Date de participation à la RD	Capacité de réduction demandée (kW))	Capacité réduite (kW)
Metal 1	18:00–19:00, 13 June 2019	8000	8777
Metal 2	17:00–20:00, 15 May 2019 16:00–17:00, 13 June 2019	24000/24000/24000 24000	25737/25874/26822 24279
Metal 3	18:00–19:00, 13 June 2019	8000	10727
Forge 1	18:00–19:00, 13 June 2019	6000	4440
Forge 2	18:00–19:00, 13 June 2019	4000	9
Steel 1	18:00–19:00, 13 June 2019	4000	3925
Steel 2	18:00–19:00, 13 June 2019	60000	195415
Cement 1	18:00–19:00, 13 June 2019	45000	51198
Cement 2	18:00–19:00, 13 June 2019	13000	18999
Paper	18:00–19:00, 13 June 2019	25000	12510

Table 3.1 : Les données de la participation des usines au programme de la réponse à la demande

3.4 Préparation des données

La première étape dans la préparation des données consiste à calculer la **Charge de Base de la Clientèle** [10], **(CBL)**. Le CBL est défini comme la consommation énergétique moyenne d'une usine pendant les périodes normales, sans intervention des programmes de réponse à la demande.

Une fois le CBL calculé, il est utilisé pour imputer les valeurs de consommation des usines pendant les périodes de réponse à la demande, en remplaçant les valeurs manquantes ou réduites par la valeur du CBL correspondante pour ces minutes. Cela permet de standardiser les données et d'obtenir une mesure plus précise de la consommation énergétique normale des usines.

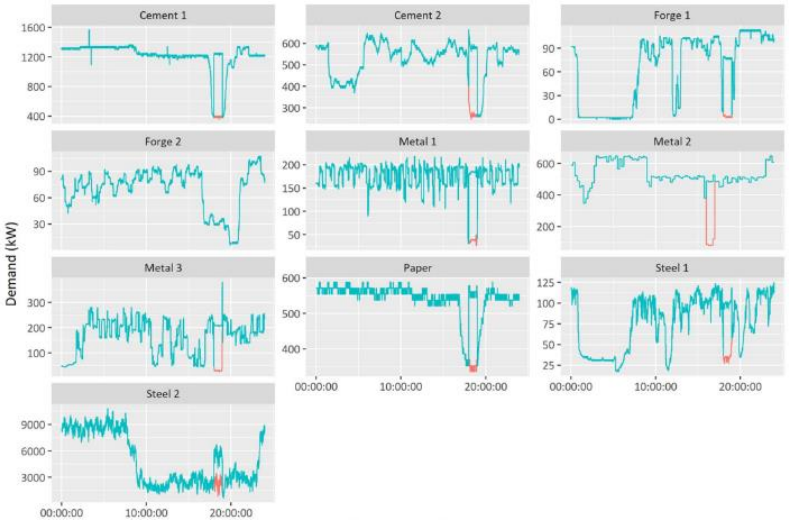

Figure 3.4 : Imputation des données de consommation des usines par les valeurs de CBL à chaque minute

3.4.1 Gestion des données manquantes

L'usine **Metal_2** contient 10,13 % de données manquantes, tandis que l'usine **Steel_2** présente 1,51 % de données manquantes.

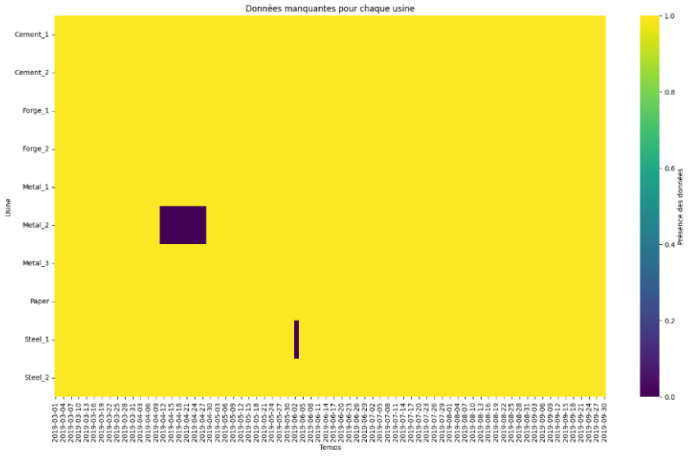

Figure 3.5 : Visualisation des données manquantes au cours du temps

Pour traiter ces données manquantes, nous avons utilisé une technique d'imputation basée sur la moyenne de la consommation selon le jour de la semaine. Cette méthode consiste à remplacer les valeurs manquantes par la moyenne des consommations observées pour le même jour de la semaine, en utilisant une table de pivotage ("pivot_table").

La technique d'imputation avec la moyenne selon le jour de la semaine est mise en œuvre comme suit :

1. **Création d'une table de pivotage** : Une table de pivotage est créée pour regrouper les données par jour de la semaine et calculer la consommation moyenne pour chaque jour.
2. **Remplacement des valeurs manquantes** : Les valeurs manquantes dans les données sont remplacées par la moyenne calculée pour le jour correspondant de la semaine. Cette méthode permet d'assurer une continuité et une cohérence dans les données, en utilisant les tendances de consommation typiques observées pour chaque jour de la semaine.

3.4.2 Nettoyage des données aberrantes

L'usine **Ciment_2** contient 4 valeurs aberrantes, l'usine **Ciment_1** en compte 38, et l'usine **Paper** a une valeur aberrante. Ces valeurs aberrantes ont été détectées à l'aide de la méthode de l'**Intervalle Interquartile (IQR)** et ont été résolues par suppression, comme illustré dans la figure ci-dessous.

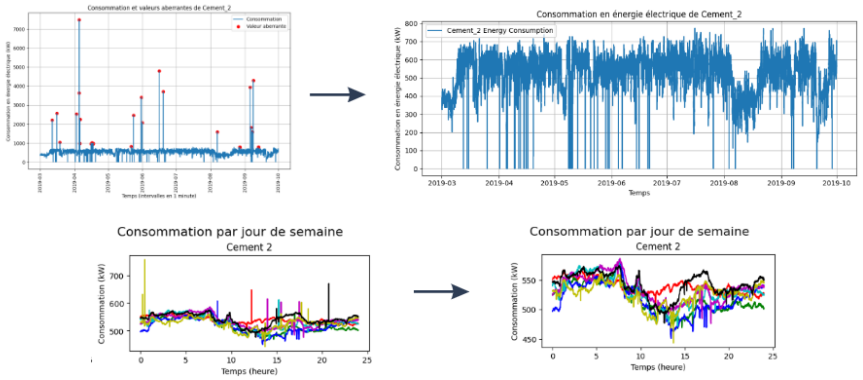

Figure 3.6 : Suppression des valeurs aberrantes

Méthode de Détection des Valeurs Aberrantes avec l'IQR :

1. **Calcul des Quartiles** : Le premier quartile (Q1) et le troisième quartile (Q3) sont calculés pour la distribution des données. Q1 est le 25e percentile et Q3 est le 75e percentile des valeurs.

2. **Calcul de l'IQR** : L'IQR est la différence entre Q3 et Q1 :

$$IQR = Q3 - Q1$$

Détermination des Seuils : Les seuils pour identifier les valeurs aberrantes sont définis en utilisant l'IQR. Les valeurs en dehors des bornes suivantes sont considérées comme des valeurs aberrantes :

$$\text{Seuil inférieur} = Q1 - 1{,}5 \times IQR$$

$$\text{Seuil supérieur} = Q3 + 1{,}5 \times IQR$$

3. **Identification des Outliers** : Les valeurs au-delà de ces seuils sont classées comme aberrantes.

4. **Suppression** : Les valeurs aberrantes détectées sont supprimées pour éviter leur impact sur l'analyse et les modèles de prédiction.

Cette méthode permet de nettoyer les données en éliminant les valeurs extrêmes qui pourraient fausser les analyses statistiques et les performances des modèles.

3.4.3 Transformation des données

Les données initiales proviennent de différentes bases de données, chacune contenant deux colonnes principales : le temps et la consommation énergétique. La transformation des données s'est déroulée en plusieurs étapes :

- **Fusion des données** :
    - Toutes les bases de données des usines ont été regroupées dans une seule base appelée **"all_data"** pour centraliser les informations et faciliter les traitements ultérieurs.

- **Répliquage des données temporelles** :
    - La colonne de temps a été décomposée en plusieurs colonnes spécifiques : **mois**, **jour**, **jour de la semaine**, **heure**, et **minute**. Ces nouvelles colonnes ont ensuite été encodées pour être utilisées dans les modèles.

	dateTime	FactoryCons	factory	month	day	hour	minute	day_of_week	Factory_code
0	2019-03-01	1164.80	Cement_1	3	1	0	0	4	0
1	2019-03-01	369.60	Cement_2	3	1	0	0	4	1
2	2019-03-01	86.40	Forge_1	3	1	0	0	4	2
3	2019-03-01	107.52	Forge_2	3	1	0	0	4	3
4	2019-03-01	184.32	Metal_1	3	1	0	0	4	4

Tableau 3.2 : Les données après décomposition

- **Normalisation de la consommation** :
    - La colonne de consommation énergétique a été normalisée à l'aide de l'outil **MinMaxScaler()** [11], qui ajuste les valeurs entre 0 et 1, ce qui permet de réduire la variance entre les différentes échelles des données et de stabiliser l'apprentissage des modèles.

- **Encodage cyclique des variables temporelles** :
    - Pour mieux représenter la nature cyclique des variables temporelles (telles que les heures et les jours), un **encodage cyclique** a été appliqué. En effet, les heures et les jours suivent un cycle (par exemple, après 23h vient 00h, ou après dimanche vient lundi), et cet encodage permet de mieux capturer ces relations.
    - L'encodage cyclique est réalisé en convertissant ces variables en deux nouvelles colonnes, **sinus** et **cosinus**, à l'aide des formules suivantes :

$$\text{sinus} = \sin\left(\frac{2\pi \times \text{variable}}{\text{max}}\right)$$

$$\text{cosinus} = \cos\left(\frac{2\pi \times \text{variable}}{\text{max}}\right)$$

	dateTime	FactoryCons	factory	month	day	hour	minute	day_of_week	Factory_code	month_sin	month_cos	day_sin	day_cos	hour_sin	hour_cos	minute_sin	minute_cos	day_of_week_sin	day_of_week_cos
0	2019-03-01	0.077828	Cement_1	3	1	0	0	4	0	1.0	1.0	0.600779	0.989739	0.5	1.0	0.5	1.0	0.277479	0.0
1	2019-03-01	0.024696	Cement_2	3	1	0	0	4	1	1.0	1.0	0.600779	0.989739	0.5	1.0	0.5	1.0	0.277479	0.0
2	2019-03-01	0.005773	Forge_1	3	1	0	0	4	2	1.0	1.0	0.600779	0.989739	0.5	1.0	0.5	1.0	0.277479	0.0
3	2019-03-01	0.007184	Forge_2	3	1	0	0	4	3	1.0	1.0	0.600779	0.989739	0.5	1.0	0.5	1.0	0.277479	0.0
4	2019-03-01	0.012316	Metal_1	3	1	0	0	4	4	1.0	1.0	0.600779	0.989739	0.5	1.0	0.5	1.0	0.277479	0.0

Tableau 3.3 : Les données après encodage cyclique

Cela permet de conserver les propriétés cycliques des données temporelles et d'améliorer les performances du modèle de prédiction.

3.5 Clustering

Afin de généraliser le projet et de le rendre applicable à toute usine de fabrication, nous regroupons les usines en fonction de leur profil de consommation par jour de la semaine. Pour déterminer les usines à regrouper ensemble, nous utilisons la méthode du coude avec l'algorithme **TimeSeriesKMeans**.

La méthode du coude permet d'identifier le nombre optimal de clusters en traçant la somme des distances intra-cluster pour différents nombres de clusters. Le graphique obtenu montre un point d'inflexion au niveau du nombre 4, ce qui indique que le nombre optimal de clusters à utiliser est 4. Ainsi, nous allons procéder au clustering avec 4 clusters pour regrouper les usines en fonction de leurs profils de consommation.

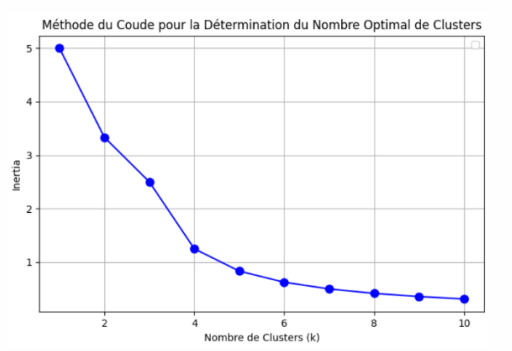

Figure 3.7 : Visualisation des résultats de la méthode du coude

On obtient 4 groupes d'usines auxquels on va appliquer les modèles de prédiction :

Cluster	**1**	**2**	**3**	**4**
Usines	Forge 1 Metal 1 Metal 3 Acier 1	Ciment 1 Ciment 2 Acier 2.	Metal 2	Forge 2 Papier

Tableau 3.4 : Les clusters obtenus

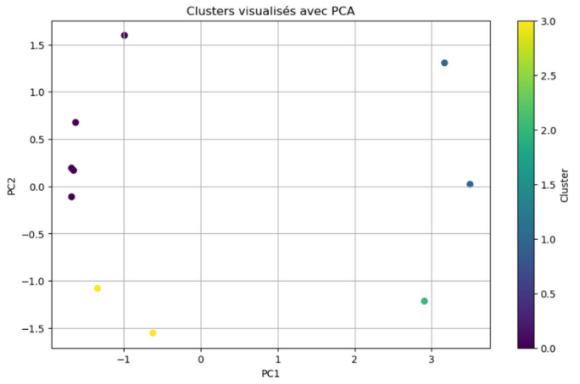

Figure 3.8 : Visualisation des clusters obtenus

3.6 Entrainement et évaluation des modèles

Nous avons testé plusieurs modèles pour chaque cluster afin de déterminer celui qui prédit le mieux la consommation énergétique des usines au sein de chaque cluster.

1/ Pour le 1er cluster :

Nous avons testé :

-Un modèle **RandomForestRegressor** avec 100 arbres de décision (n_estimators=100). Chaque arbre a une profondeur maximale de 10 niveaux (max_depth=10), ce qui aide à contrôler la complexité et éviter le surapprentissage. Le paramètre random_state=42 a été fixé pour garantir la reproductibilité des résultats.

-Un modèle **XGBRegressor** avec 1000 arbres de décision (n_estimators=1000) et un taux d'apprentissage de 0.01 (learning_rate=0.01) pour des mises à jour précises des poids. La profondeur maximale des arbres - caractéristiques (colsample_bytree=0.8) sont utilisés pour chaque arbre, ce qui aide à améliorer la performance et la généralisation du modèle.

- un modèle **LSTM** simple avec une architecture composée de 50 unités LSTM. La fonction d'activation utilisée est la tanh, et la régularisation est assurée par un dropout de 0.2 pour éviter le surapprentissage. Le modèle est optimisé avec l'algorithme d'optimisation **Adam** [12] pour une convergence efficace.

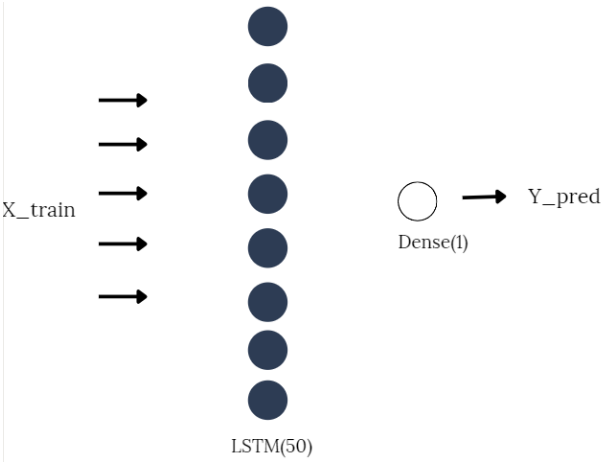

Figure 3.9 : Illustration du modèle à une couche LSTM

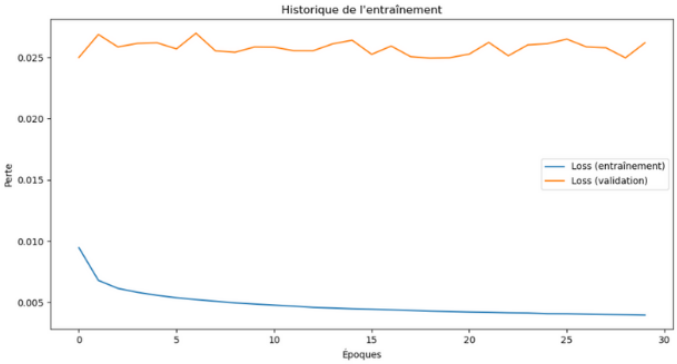

Figure 3.10 : Historique d'entrainement du modèle LSTM

- un modèle **LSTM** séquentiel avec quatre couches LSTM. La première couche contient 128 unités avec une activation **tanh** et retourne des séquences (return_sequences=True). Les couches suivantes sont configurées avec 64, 32 et 16 unités respectivement, toutes utilisant l'activation **tanh**. À chaque étape, un dropout de 0.3 est appliqué pour éviter le surapprentissage. Le modèle se termine par une couche dense avec une unité de sortie pour la prédiction.

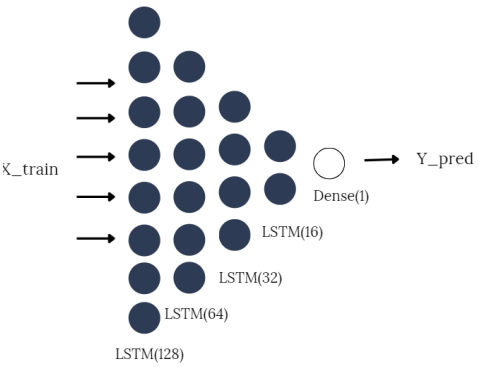

Figure 3.11 : Illustration du modèle à 4 couches LSTM

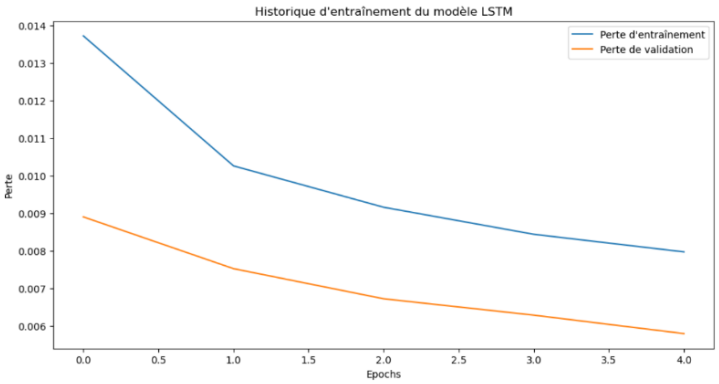

Figure 3.12 : Historique d'entrainement du modèle à 4 couches LSTM

Modèle	Random Forest Regressor	XGBoost	LSTM(1)	LSTM(4)
Coefficient de détermination	79.96%	91.6%	37.02%	83.05%
EQM	0.0068	0.0029	0.0491	0.0058
REQM	0.0827	0.0536	0.2215	0.0671

Tableau 3.5 : Evaluation des modèles pour Cluster 1

En se basant sur les métriques que nous avons expliqué dans le chapitre précédent, le meilleur modèle en termes de performances pour Cluster 1 est XGBoost.

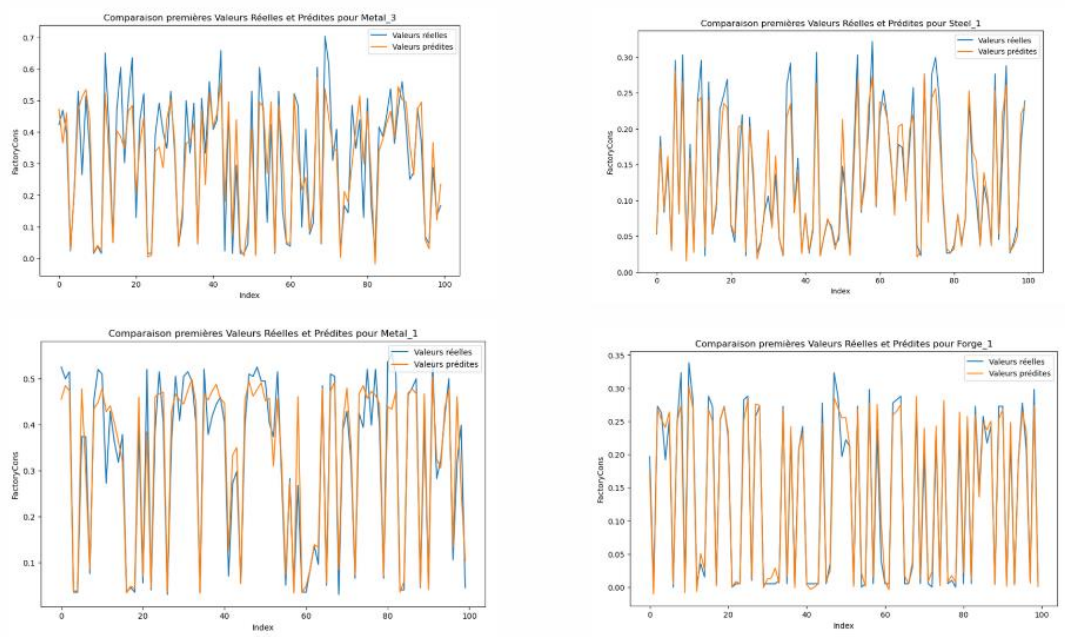

Figure 3.13 : Visualisation des résultats de XGBoost pour Cluster 1

2/ Pour le 2éme cluster :

Nous avons testé :

- Un modèle Random Forest Regressor comme celui testé pour le cluster précédent.

- Un modèle XGBoost comme celui testé pour le cluster précédent.

- Un modèle LSTM séquentiel à quatre couches comme celui testé pour le cluster précédent.

- Un modèle **GRU** séquentiel composé de quatre couches GRU. La première couche contient 128 unités avec une activation **tanh** et retourne des séquences (return_sequences=True). Les couches suivantes sont configurées avec 64, 32, et 16 unités respectivement, toutes activées par **tanh**. Chaque couche est suivie d'un dropout de 0.3 pour réduire le risque de surapprentissage. Enfin, une couche dense avec une seule unité est utilisée pour la prédiction finale.

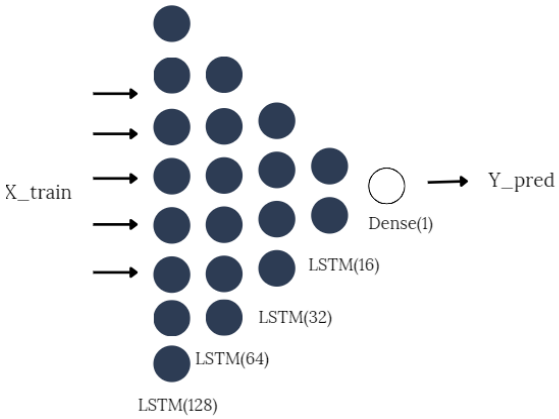

Figure 3.14 : Illustration du modèle GRU

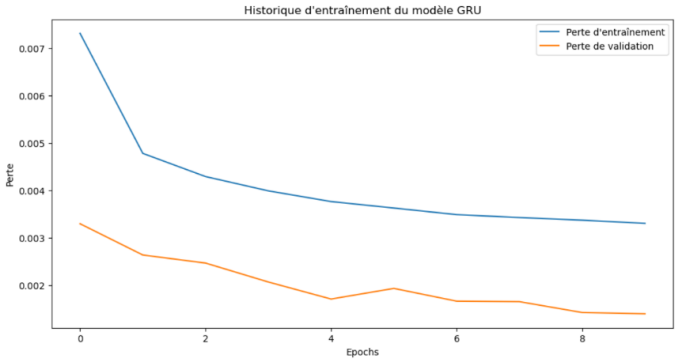

Figure 3.15 : Historique d'entrainement du modèle GRU

Modèle	Random Forest Regressor	XGBoost	LSTM(4)	GRU(4)
Coefficient de détermination	97.26%	98.77%	97.88%	97.38%
EQM	0.0015	0.0007	0.0011	0.0014
REQM	0.0384	0.0257	0.0338	0.0375

Tableau 3.6 : Evaluation des modèles pour Cluster 2

En se basant sur les métriques que nous avons expliqué dans le chapitre précédent, le meilleur modèle en termes de performances pour Cluster 2 est XGBoost.

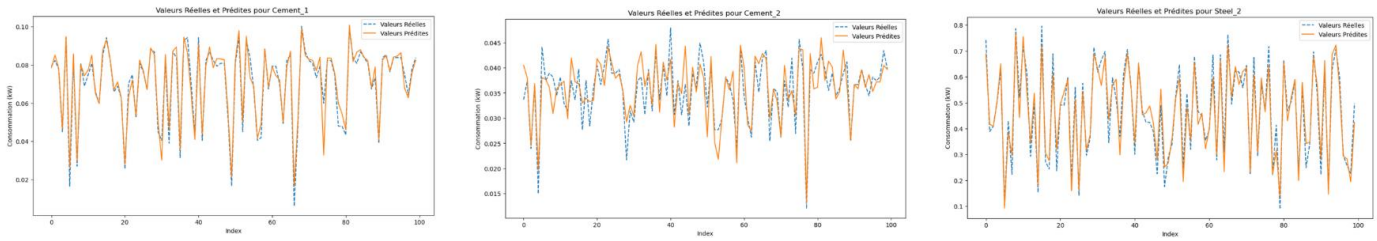

Figure 3 .16 : Visualisation des résultats de XGBoost pour Cluster 2

3/ Pour le 3éme cluster :

Nous avons testé :

- Un modèle Random Forest Regressor comme celui testé pour les clusters précédents.

- Un modèle XGBoost comme celui testé pour les clusters précédents.

- Un modèle LSTM séquentiel à quatre couches comme celui testé pour les clusters précédents.

- Un modèle **GRU** séquentiel composé de quatre couches GRU celui testé pour le cluster précédent.

Modèle	Random Forest Regressor	Gradient Boosting Regressor	XGBoost	LSTM(4)	GRU(4)
Coefficient de détermination	92.81%	95.47%	95.75%	92.41%	91.48%
EQM	0.006	0.0038	0.0036	0.0064	0.0071
REQM	0.0776	0.0615	0.0596	0.0797	0.0844

Tableau 3.7 : Evaluation des modèles pour cluster 3

En se basant sur les métriques que nous avons expliqué dans le chapitre précédent, les meilleur modèle en termes de performances pour Cluster 3 est XGBoost.

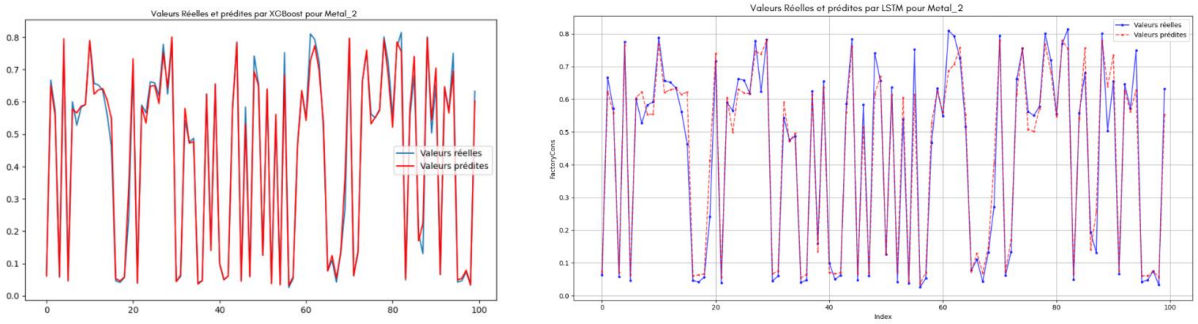

Figure 3.17 : Visualisation des valeurs réelles et prédites par XGBoost et LSTM

4/ Pour le 4éme cluster :

Nous avons testé :

- Un modèle Random Forest Regressor comme celui testé pour les clusters précédents.

- Un modèle XGBoost comme celui testé pour les clusters précédents.

- Un modèle LSTM séquentiel à quatre couches comme celui testé pour les clusters précédents.

- Un modèle **GRU** séquentiel composé de quatre couches GRU celui testé pour le cluster précédent.

Modèle	Random Forest Regressor	XGBoost	LSTM(4)	GRU(4)
Coefficient de détermination	99.44%	99.82%	99.18%	97.38%
EQM	0.0007	0.0002	0.0010	0.0014
REQM	0.0261	0.0149	0.0317	0.0375

Tableau 3.8 : Evaluation des modèles pour Cluster 4

En se basant sur les métriques que nous avons expliqué dans le chapitre précédent, les meilleur modèle en termes de performances pour Cluster 3 est XGBoost.

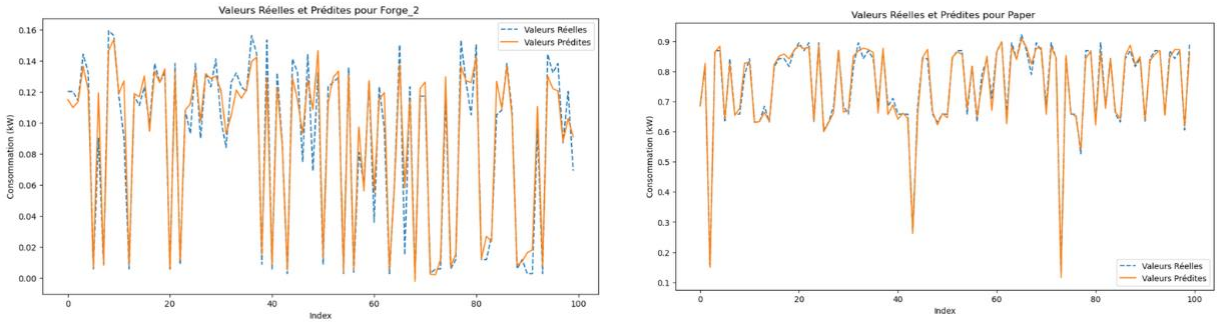

Figure 3.18 : Visualisation des valeurs réelles et prédites par XGBoost pour Cluster 4

3.7 Conclusion

Ce chapitre est consacré à la présentation des outils utilisés et des travaux réalisés au cours et aux résultats obtenus tout au long de la période du stage.

Conclusion générale

Tout au long de ce stage, nous avons développé un modèle de prédiction de la consommation d'énergie électrique des usines de fabrication, en nous basant sur des données collectées à intervalles réguliers. Le projet a débuté par une phase de préparation des données, comprenant le nettoyage des données manquantes, l'imputation et la gestion des valeurs aberrantes. Ensuite, nous avons procédé à la transformation des données, incluant l'encodage cyclique des variables temporelles. Une étape de clustering a été effectuée pour regrouper les usines selon leurs profils de consommation, puis des modèles de régression ont été testés pour chaque cluster afin d'obtenir les prédictions les plus précises. Nous avons utilisé plusieurs algorithmes avancés, tels que RandomForest, XGBoost, LSTM, et GRU, pour déterminer le modèle le plus performant.

Ce stage m'a permis d'acquérir des compétences techniques solides en matière de traitement de données, de modélisation et de machine learning, tout en approfondissant ma maîtrise des outils comme Python, TensorFlow, et scikit-learn. J'ai également appris à utiliser des techniques avancées de manipulation des données et de modélisation pour des problématiques complexes, comme la prédiction énergétique. Cette expérience m'a permis de développer un esprit critique sur le choix des méthodes et algorithmes, et de mieux comprendre l'importance d'une bonne préparation des données pour obtenir des résultats pertinents et robustes.

Bibliographie

1. https://www.sagemcom.com/fr/qui-sommes-nous?language_content_entity=fr
2. https://fr.wikipedia.org/wiki/R%C3%A9ponse_%C3%A0_la_demande
3. https://levelup.gitconnected.com/unveiling-patterns-in-time-a-guide-to-time-series-clustering-with-tslearn-50a2ff305afe
4. https://towardsdatascience.com/random-forest-regression-5f605132d19d
5. https://medium.com/the-modern-scientist/gradient-boosting-regressor-the-best-machine-learning-algorithm-370b7b41ad09
6. https://blent.ai/blog/a/xgboost-tout-comprendre
7. https://datavalue-consulting.com/deep-learning-reseaux-neurones-recurrents-rnn/
8. https://www.datasciencetoday.net/index.php/fr/machine-learning/148-reseaux-neuronaux-recurrents-et-lstm#:~:text=Les%20r%C3%A9seaux%20de%20longue%20m%C3%A9moire,tr%C3%A8s%20longs%20entre%20les%20deux.
9. https://medium.com/@anishnama20/understanding-gated-recurrent-unit-gru-in-deep-learning-2e54923f3e2
10. https://www.sciencedirect.com/science/article/pii/S2666412723000053#:~:text=In%20DR%20programs%2C%20customer%20baseline,the%20efficiency%20of%20these%20programs.
11. https://pyihub.org/sklearn-minmaxscaler/
12. https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/