UNIVERSITÉ DE STRASBOURG

Capgemini engineering

Rapport de stage de fin d'études

Etudiant :
Quentin Dumont

Encadrant :
Ufuk Halisdemir

Promotion 2021 - 2022

Remerciement

Je tiens à remercier mon maître de stage Ufuk Halisdemir, Chef de projet et de recherche, dont son soutien et son support lors de mes 6 mois de stage au sein de Capgemini Engineering ont été d'une aide précieuse pour mon apprentissage. Je remercie également mes camarades Matthias Foyer, étudiant en Master 2 de Sciences des données et systèmes complexes et Thomas Blondel, étudiant en alternance dont une collaboration s'est faite lors d'un projet et qui a été très enrichissante.

Enfin, je désire remercier mes professeurs de l'université de Strasbourg du master CSMI, qui m'ont fourni les connaissances nécessaires au bon déroulement de mon stage, mais également de nombreux conseils lors de mes études.

Table des matières

1 Introduction

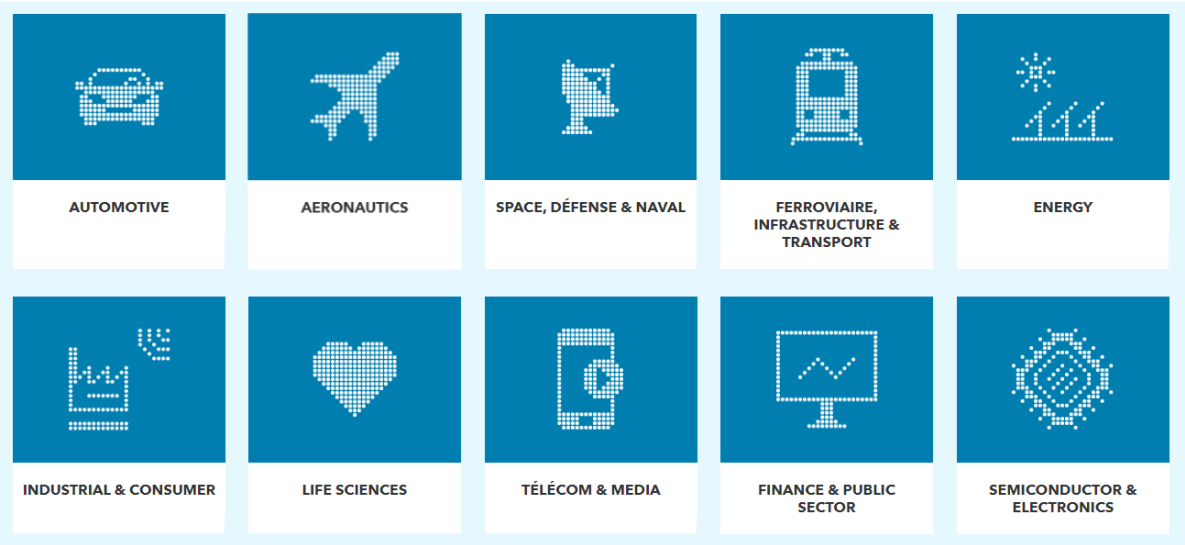
Dans le cadre de la recherche et développement de l'entreprise Capgemini Engineering, j'ai effectué un stage du 1 février 2022 au 31 juillet 2022 dans lequel j'ai pu m'orienter dans le secteur de l'énergie à travers deux différents projets. Étant sous la supervision de mon maître de stage Ufuk Halisdemir, Chef de projet de recherche, j'ai pu bénéficier d'un suivi et soutien constant ainsi que la possibilité d'effectuer mon stage sous d'excellentes conditions.

J'ai pu ainsi à travers ce stage appréhender différents aspects des méthodes et techniques de développement dans les industries dont l'énergie est un enjeu majeur et qui s'inscrit ainsi dans la continuité du domaine de mes projets antérieurs effectués lors de mes études. Mon stage au sein du secteur de l'énergie du département R&D d'Altran situé à Illkirch-Graffenstaden a consisté essentiellement à l'étude de deux projets dont le premier se base sur la poursuite des recherches d'optimisation de la récupération de la chaleur fatale d'un procédé industrielle dans le cadre du projet Anagreen, et dont le second consiste à obtenir des prévisions sur un horizon de temps court de l'irradiation solaire de surface en France métropolitaine à travers la construction d'une interface graphique. Au-delà d'enrichir et d'approfondir mes connaissances dans le domaine de l'optimisation et de la science des données, j'ai été amené à comprendre dans quelle mesure la recherche et les missions sont effectués, et ce, au cœur d'un groupe leader mondial de service d'ingénierie.

En vue de fournir un rapport détaillé des 6 mois accomplis au sein de la société Capgemini Engineering, ce rapport présente en premier lieu l'entreprise ainsi que les objectifs de mon stage, puis d'apporter en deux parties, les contextes et les missions que j'ai pu accomplir lors de ce dernier, chacune présentant un des deux sujets de stage. Enfin, une conclusion clôtura ce rapport sur les résultats obtenus et les nombreux apports que j'ai pu obtenir.

1.1 Présentation de l'entreprise

Capgemini Engineering est un leader mondial de conseil en ingénierie fondée en France en 1982 et est la marque du groupe Capgemini qui réunit les services d'ingénierie et de recherche et développement. Elle est associée à l'ancienne société Altran rachetée en 2020 par le groupe Capgemini à la suite d'une opération d'acquisition de parts, ce qui a conduit à sa nouvelle dénomination "Capgemini Engineering". Elle accompagne grâce à son expertise dans les dernières technologies digitales ou logicielles ainsi que ses nombreuses recherches de nombreux clients ayant pour objectif d'accélérer leurs développements dans leurs entreprises ou industries opérant dans différents secteurs dont notamment l'aéronautique, spatial, l'énergie, les sciences de la vie, l'automobile ou biens d'autres.



De plus, fort de sa notoriété et de son engagement, Capgemini Engineering emploie 52 000 ingénieurs et scientifiques dans plus de 30 pays répartis dans le monde et a réalisé un chiffre d'affaires de 18 milliards d'euros en 2021 ce qui en fait le groupe leader mondial de conseil. Elle est également amenée à devenir partenaire de plusieurs entreprises pour répondre à l'ensemble des demandes et missions qui lui sont confiées et prône pour un avenir durable guidé par des technologies toujours plus innovantes.

1.2 Projets

Au cours de mon stage, deux projets m'ont été proposés et répartis de manière équitable, soit d'une durée de 3 mois chacun. Le premier projet consistait à poursuivre les travaux effectués par les précédents stagiaires afin d'apporter de nouvelles solutions. Ces travaux ont été basés sur l'optimisation de la récupération de la chaleur fatale d'un procédé industriel intitulé MidRex à travers des méthodes thermodynamiques et de segmentation de série temporelle représentant l'évolution des flux dans le procédé au cours du temps. Cependant, une nouvelle approche se basant sur une étape de "clustering" afin d'obtenir de nouveau résultat a été soutenue et m'a été confié pour être réalisée. L'ensemble des méthodologies et des résultats obtenus portant sur ce sujet sont détaillés dans la section 2.

Par la suite et dès lors que le premier projet fut clos, j'ai pu basculer sur le second sujet et rejoindre ainsi un étudiant de Master 2 dans l'étude de l'inférence de l'irradiation solaire en France. J'ai eu pour objectif d'apporter d'une part un module de prévision de l'irradiation solaire de surface sur un horizon de temps court, et ce, sur l'ensemble du territoire français, et d'autre part d'aider à la construction d'une interface graphique hébergée sur un navigateur web avec l'aide notamment d'un étudiant en alternance dans le développement web. De même que pour le précédent projet, l'ensemble des méthodologies et des résultats obtenus sont détaillés dans la section section 3.

2 Projet Anagreen

2.1 Contexte :

En France, le secteur de l'industrie se positionne à la troisième place des secteurs d'activités consommant le plus d'énergie finale, comme le montre la Fig. 1 suivante :

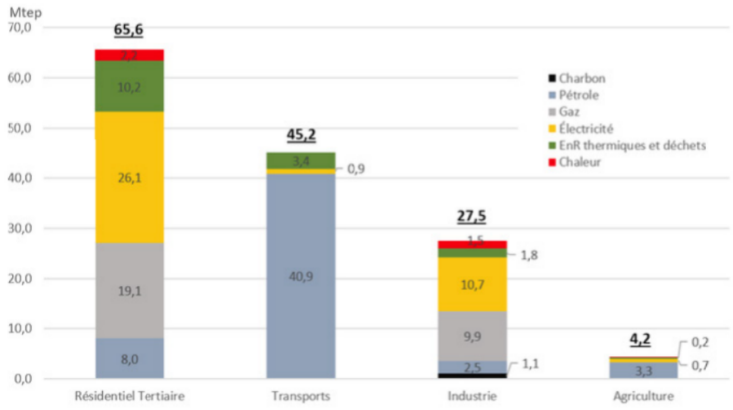


FIGURE 1 – Panorama de la consommation d'énergie en France en 2018 (source : MTES [7])

Elle représente à elle seule 19 % de la consommation d'énergie nationale dont ses principales sources d'énergies consommées par ce secteur sont **les énergies fossiles** qui représentent **56%** et **l'énergie électrique** qui représente **32%**. Néanmoins, ces chiffres sont nuancés lorsque nous nous intéressons principalement à la branche des industries destinées à la sidérurgie, à la fonderie et au travail des métaux, comme

détaillé sur la Fig. 2 suivante qui présente la répartition de la consommation d'énergie dans les différents secteurs de l'industrie.

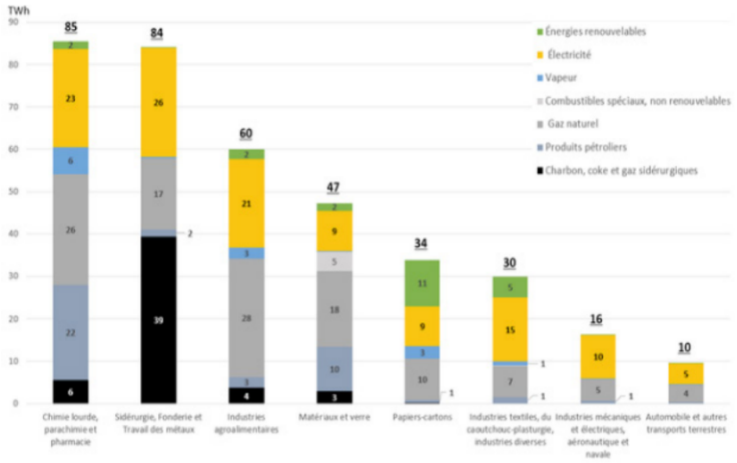


FIGURE 2 – Répartition de la consommation d'énergie dans l'industrie par sous-secteur et par énergie en 2016 (Source : Ceren [5])

Nous observons que les industries destinées à la sidérurgie, à la fonderie et au travail des métaux consomment **30% d'énergie électrique** contre **69% d'énergies fossiles** et qui servent principalement à produire de la chaleur. Cependant, seulement une faible partie, soit `20%-40%` de cette chaleur produite est utilisée lors de la production, ce qui implique que la quantité restante de la chaleur non consommée (que l'on nomme la chaleur fatale) lors des procédés industriels représente `60%-80%` de la chaleur. Nous pouvons alors constater le fort intérêt sur un plan économique et environnemental de récupérer cette chaleur fatale présente sous forme liquide, gazeuse ou diffuse. En effet, en redirigeant cette chaleur fatale dans un conduit de réseau afin de satisfaire par exemple les besoins en énergie de l'entreprise ou simplement de la réimplanter dans le réseau urbain, nous pouvons réduire considérablement les coûts liés à la consommation d'énergie, mais également de réduire les émissions de gaz à effet de serre. Cependant, une partie de la chaleur fatale est définitivement irrécupérable. C'est ainsi, dans ce contexte très précis que le **projet Anagreen** a été mis en œuvre ayant pour objectif d'apporter des solutions innovantes et générales afin de récupérer l'énergie fatale tout en associant les coûts d'une telle implémentation ainsi que l'estimation des gains en retour afin d'aider les industries dans leurs choix de développement.

2.2 Anagreen

La récupération de la chaleur fatale est un enjeu essentiel pour l'environnement, la performance des industries et économique, en particulier lorsque les industries sont très consommatrices en énergie telle que par exemple les industries sidérurgiques. En effet, l'ADEME a estimé que la quantité de chaleur fatale supérieure à 100°C représente $51TWh$, soit `10%` de la production. Ainsi, dans ce contexte a eu lieu en 2019 l'apparition du projet **Anagreen** [4] ayant pour objectif de renforcer et d'élargir les missions de diagnostic énergétique dans les industries ainsi que d'améliorer les méthodes de récupération de la chaleur fatale afin d'avoir de meilleures performances et de meilleures estimations de coûts et des retours sur l'investissement. Afin de consolider ce projet, un partenariat a été mis en place entre ArcelorMittal (site industriel), Altran (Capgemini Engineering) et l'ADEME qui finance la moitié du montant total du projet.

L'approche de ce projet est d'intégrer et de combiner les différentes équipes nommées "Work packages" spécialisées dans les domaines suivants afin d'atteindre notre objectif :

- Équipe Data science : Exploiter les données d'enregistrements de capteurs industriels afin d'identifier et fournir des intervalles de régime et les variations de chaleur potentiellement récupérable.

- Équipe thermique : Récupère les résultats obtenus par l'équipe Data science afin de construire des modèles d'intégration thermique. Ces modèles vont être appliqués sur des intervalles de temps spécifique afin de concevoir des réseaux d'échange optimaux selon les contraintes et les souhaits de l'industriel.
- Équipe Digital : Crée une plateforme web afin d'analyser, de visualiser et de présenter les résultats obtenus. Elle est paramétrable selon le cas d'étude souhaité afin de faciliter la décision de l'investisseur.
- Équipe ArcelorMittal : Fournit les données, les cas d'étude et apporte une expertise métier.

2.3 État des lieux à mon arrivée

Lors de mon arrivée, de nombreuses avancées ont été faites dans la partie data science et thermique dont les principaux travaux concernant l'étude du projet sont :

☐ Data science :
- Traitement des jeux de données (nettoyage, analyse, etc ...)
- Segmentation des séries temporelles représentant les flux univariés et multivariés.
- Regroupement et détection des régimes.
- Bibliographie

☐ Thermique :
- Analyse et application de la méthode de pincement
- Construction des modèles thermique
- Mis en place d'une méthode de réseau d'échangeur optimal nommé `MER` obtenu en appliquant la méthode d'analyse par pincement
- Bibliographie

Ces travaux ont tous été implémentés sous Python avec des "notebooks" fournis afin de faciliter la compréhension des études effectuées. Étant situé entre l'équipe thermique et data science sur ce projet, j'ai ainsi commencé par reprendre entièrement les travaux et les algorithmes existants développés afin de me les approprier et de poursuivre leurs développements.

2.4 Objectif et Roadmap :

L'objectif de mon stage sur ce sujet était de poursuivre les algorithmes implémentés par les précédentes équipes dont les études ont été basées sur l'optimisation de la récupération de la chaleur fatale d'un procédé industriel intitulé MidRex à travers des méthodes thermodynamiques (méthode de pincement) et de la segmentation de série temporelle représentant l'évolution des flux dans le procédé au cours du temps. Enfin, une nouvelle approche m'a été confiée afin d'obtenir de nouvelles solutions en développant des algorithmes de "clustering" sur les séries temporelles et d'appliquer les méthodes de pincements à travers un modèle thermodynamique pour proposer des solutions de réseau d'échangeur optimal évaluées par différentes valeurs énergétiques et financières [1].
Afin de structurer et d'organiser les nombreuses tâches de ce projet, une **roadmap** est mise en place et présente l'ensemble des étapes afin d'atteindre l'objectif présenté ci-dessus :

☐ Étape 1 : Reprendre entièrement les travaux et les algorithmes existants.

☐ Étape 2 : Dresser un état de l'art.

☐ Étape 3 : Améliorer et optimiser les algorithmes de clustering implémentés.

☐ Étape 4 : Développer l'algorithme TICC qui consiste à segmenter et regrouper les segments des séries temporelles.

1. les différentes valeurs énergétiques et financières sont : MER, CAPEX (coût d'investissement), l'OPEX (coût d'exploitation), etc ...

☐ Étape 5 : Appliquer la méthode de pincement à travers l'utilisation d'une boite noire pour construire un réseau optimal de revalorisation de la chaleur fatale grâce aux résultats obtenus.

☐ Étape 6 : Développer un algorithme de front de Pareto afin de positionner de manière optimale les échangeurs selon les différentes valeurs énergétiques et financières et de proposer différentes solutions selon les exigences des clients.

2.5 Méthodes et approches :

Améliorer les performances énergétiques des procédés industriels nécessite des méthodes d'optimisation systématique et rigoureuse. Les premières méthodes datent des années 1970 et sont basées sur la récupération de la chaleur fatale dans l'industrie avec une méthode de pincement développée par Linnhoff & al. La méthode de pincement est une méthode basée sur des lois thermodynamiques pour identifier les possibles échanges de chaleur entre les procédés. L'objectif est de réduire l'utilisation des utilités chaudes et froides fournissant l'énergie nécessaire à la production afin de réaliser des économies d'énergie et financière, mais également de réduire les émissions de gaz polluant. Cette méthode fournit une procédure systématique pour créer des réseaux d'échangeurs de chaleur pour récupérer la chaleur fatale. Historiquement, l'intégration énergétique et plus particulièrement la récupération de la chaleur fatale a été principalement appliquée sur des méthodes statiques consistant simplement à conserver les valeurs moyennes des quantités physiques sur l'ensemble des flux au cours du temps. Les méthodes quant à elles consistant à découper le procédé continu en un procédé discontinu isolant les quantités physiques variant au cours du temps ont été ignorées car considérées trop complexe. Cependant, il a été démontré que plusieurs opportunités d'économie d'énergie et économique sont possibles à travers cette dernière. Depuis que les méthodes discontinues ou dynamiques ont reçu moins de considération que les méthodes statiques, elles ont attiré de nombreuses recherches ces dernières années. Étendre l'analyse de la méthode de pincement en prenant en compte les variations dans le temps des flux intervenant dans le procédé est un atout dans le développement de la méthode de pincement. Le "Time slice Model" (TSM) est une première approche dans le développement de la méthode, elle consiste à segmenter la série temporelle de notre procédé en plusieurs intervalles de temps réguliers dans lesquelles les quantités physiques peuvent être considérées constantes Fig. 3 (b). La méthode pincement standard Fig. 3 (a) est ensuite appliquée sur chacun de ces intervalles de temps. Cette méthode bien que simpliste, permet d'identifier plusieurs améliorations.

- Identifie les échanges directs entre les flux lorsqu'ils existent conjointement. Un réseau d'échangeur est créé sur chaque intervalle de temps. Le réseau final est la somme des réseaux optimisés sur chacun de ces intervalles.
- Reprogramme les flux pour maximiser les possibles échanges de flux direct. Cette approche est limitée par les contraintes associées aux procédés spécifiques.
- Utilise des technologies de stockage de chaleur pour maximiser la récupération de la chaleur fatale, cependant, ce type de solution nécessite une mise en place de plusieurs éléments complexes et coûteux.

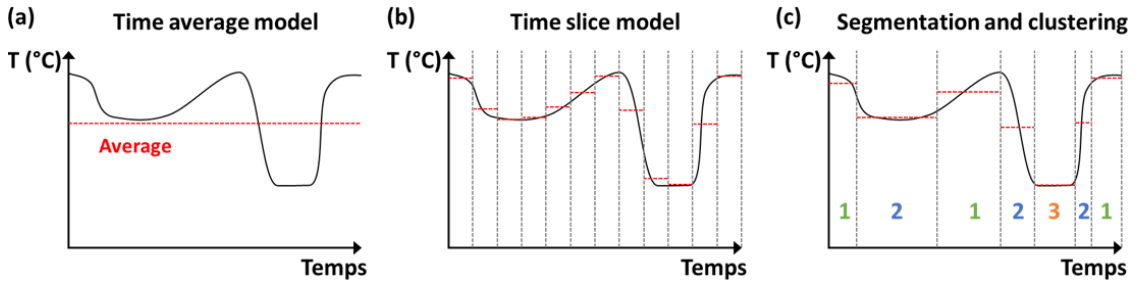


FIGURE 3 – donnée des capteurs du site industriel. (a) La méthode standard de la méthode de pincement avec les flux représentant les quantités physiques, (b) Le Time slice Model (TSM) découpant la série temporelle en plusieurs intervalles de temps réguliers et (c) l'approche proposée par Anagreen dont l'objectif est de développer un algorithme automatique pour segmenter et regrouper ces intervalles de temps qui semblent similaires.

Une estimation du coût est essentielle pour l'implémentation d'un projet de rénovation ou de la création de réseau d'échangeur. Puisque le coût est directement lié à la quantité de chaleur récupérable, il est nécessaire d'estimer le potentiel de récupération de chaleur avec précision. Dans le cas des procédés dans lesquels les quantités physiques varient dans le temps et au sein de la structure TSM Fig. 3 (b), il aurait été nécessaire de segmenter avec des intervalles de temps très courts pour pouvoir calculer ce potentiel avec précision. Cependant, cette approche est trop coûteuse en termes de temps de calcul et puisque le réseau final est la somme des réseaux optimisés obtenus sur chaque intervalle de temps, cette solution serait trop complexe pour être réalisable. L'idée est alors d'utiliser les données des capteurs avec des algorithmes non supervisés pour améliorer le "Time slice Model". Comme illustré sur la Fig. 3 (c), une approche plus sophistiquée consiste à réduire le nombre d'intervalles de temps en segmentant en premier lieu avec une segmentation adaptative, réduisant ainsi la complexité et le coût des solutions. Puis, en complément, nous proposons de regrouper ces segments similaires afin de réduire encore le nombre d'intervalles de temps et d'appliquer sur chacun d'eux, une seule fois la méthode de pincement, améliorant ainsi encore les solutions des réseaux d'échangeurs de chaleur par rapport au coût et à la complexité. réseaux d'échangeurs.

Cette section est organisée de la manière suivante, dans la partie Sect. 2.5.1 nous introduisons brièvement notre cas d'étude étant le procédé MidRex ainsi que les données utilisées, puis nous détaillerons dans la Sect. 2.5.2 la méthode de pincement qui permet de définir la cible énergétique et de créer un réseau d'échangeurs de chaleur optimisé.

2.5.1 Procédé MidRex :

Le cas d'étude est le procédé MidRex qui est un procédé sidérurgique dont l'usine est située au Canada et qui permet l'obtention de fer à partir de minerai de fer par réduction des oxydes de fer sans fusion du métal. La figure Fig. 4 présente le fonctionnement du procédé MidRex :

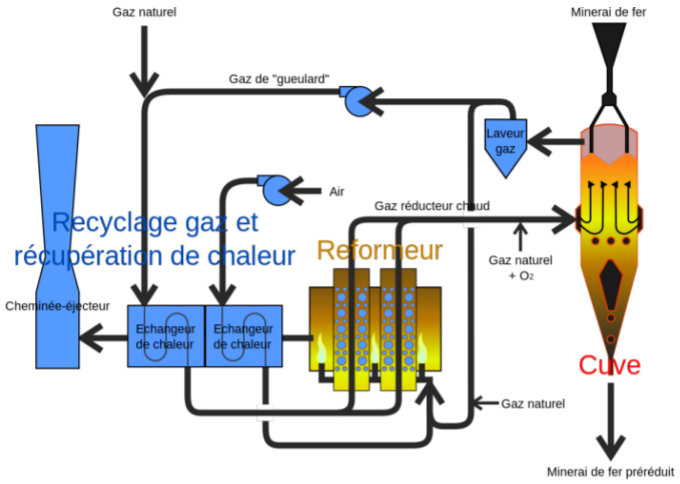


FIGURE 4 – Procédé MidRex

Explication :

La cuve sur la Fig. 4 permet la réduction entre l'oxyde de fer et le gaz réducteur. Cette réaction produit un gaz (Top gas) passant dans l'épurateur de gaz afin d'épurer ce dernier. Il va être ensuite reconduit en deux parties :

- une partie vers un réformeur.
- l'autre partie vers un compresseur avant d'ajouter un gaz naturel.

Ces deux parties sont ensuite reconduites dans un échangeur de chaleur afin de chauffer ce mélange qui sera envoyé vers l'autre reformeur. Un autre procédé non présenté sur le graphique met en place au niveau de

la cuve une boucle de refroidissement où l'on trouve un épurateur et un compresseur exploitant un gaz de refroidissement (`cooling gas`) afin de la réintroduire dans la cuve. Nous pouvons ainsi obtenir l'ensemble des flux agissant dans le procédé et enregistré à l'aide de plusieurs capteurs présents dans ce dernier. Ces enregistrements ont été rééchantillonés afin d'avoir un pas constant et sont complétés à l'aide de donnée simulée d'une longueur de 6 semaines. Les différents flux enregistrés par les capteurs sont présentés dans la figure Fig. 5 :

Flux chauds	Gaz de Refroidissement : **Cool**
	Eau de refroidissement du compresseur du top gas: **CompTop**
	Eau de refroidissement du compresseur du cooling gas : **CompCool**
	Gaz d'échappement du reformer A : **Echap_A**
	Gaz d'échappement du reformer B: **Echap B**
Flux froids	Apport de gaz naturel au reformer A : **GN21_A**
	Apport de gaz naturel au reformer B : **GN21_B**
	Apport de gaz naturel au reducing gas: **GN8**
	Apport d'air au reducing gas : **Air8**
	Gaz combustible du reformer A : **Fuel_A**
	Gaz combustible du reformer B : **Fuel_B**
	Apport de gaz naturel au Fuel_A: **GN13_A**
	Apport de gaz naturel au Fuel_B: **GNaux_B**
	Apport auxiliaire de gaz naturel au Fuel_A: **GNaux_A**
	auxiliaire de gaz naturel au Fuel_B: **GN13_B**
	Apport auxiliaire d'air au Fuel_A : **AirAux_A**
	Apport auxiliaire d'air au Fuel_B : **AirAux_B**

FIGURE 5 – Flux intervenant dans le procédé MidRex

Cette étape de récupération des données est cruciale afin de pouvoir faire un bilan de l'ensemble des flux intervenant dans le procédé et pouvoir analyser la quantité de chaleur récupérable. Les données provenant des capteurs sont ensuite traitées afin de pouvoir les utiliser dans un format adéquat pour le bon fonctionnement des algorithmes tels que la segmentation et le clustering. Un bref résumé est présenté ci-dessous :

- Nettoyage des données (valeur aberrante, donnée manquante, ...) et ré-échantillonnage avec un pas constant d'une heure.
- Normaliser les données pour les algorithmes de clustering.

Ces données proviennent ainsi de 238 capteurs positionnés à l'intérieur des équipements du procédé et mesure la température d'entrée et de sortie, le débit, la pression, la quantité de gaz etc... . L'ensemble des données couvre une période de 10 mois (janvier à novembre 2018) dont seulement la période du 16 janvier au 18 février est exploitable due à des données essentielles manquantes. Un flux est associé à 3 capteurs mesurant respectivement la température d'entrée, la température de sortie et le débit correspondant et 17 flux sont présents dans le procédé MidRex comme présenté sur la figure Fig. 5. De plus, une étape de lissage des données a été appliquée comme suggérée dans l'article de Spiegel et al [26]. La figure Fig. 6 suivante illustre une partie de la série temporelle avec les données normalisées et nettoyées de notre procédé.

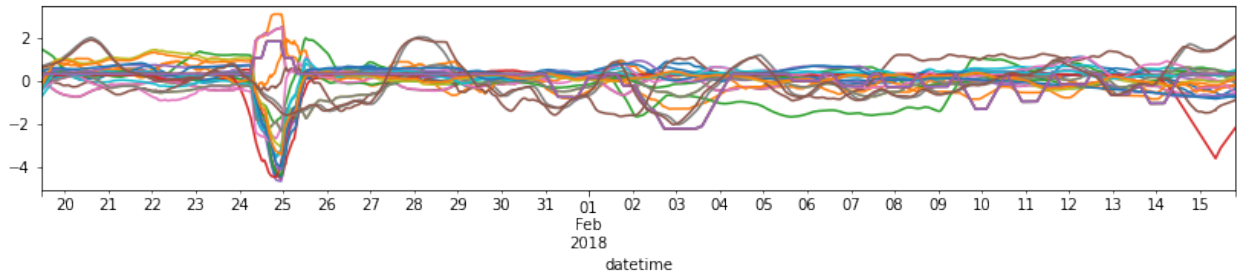


FIGURE 6 – Période couvrant le 19 janvier au 16 février de la série temporelle avec des données normalisées et nettoyées du procédé MidRex

2.5.2 Intégration énergétique

L'intégration énergétique [14] est une approche rassemblant l'ensemble des méthodologies d'optimisation et d'analyse des procédés afin de réduire de façon optimal la consommation d'énergie ou de matière première. Un bilan de récupération d'énergie d'un procédé industriel est mis en place tout d'abord en analysant et en classifiant tous les flux (chaud [2] ou froid [3]) intervenant dans le procédé. Ce bilan permet ainsi d'appliquer les méthodes d'intégration énergétique pour connaître le potentiel de récupération d'énergie entre ces différents flux. Parmi l'ensemble des méthodes que comptent l'intégration énergétique, la méthode de pincement conçue et développée par Linnhoff & al [15] dans les 1970 a été retenue puisqu'elle permet efficacement d'identifier les possibles échanges de chaleur entre les procédés. Nous présentons dans cette section une brève explication de son fonctionnement.

Méthode de pincement : .

Il s'agit de l'approche traditionnelle et la plus utilisée de l'intégration énergétique des procédés [14]. Elle permet d'optimiser notre efficacité énergétique en introduisant des procédés de récupération de la chaleur en déterminant les meilleurs réseaux d'échanges et systèmes d'utilités.

Cette méthode est basée sur des principes thermodynamiques et analyse les potentiels échanges de chaleur entre les flux froids et les flux chauds de façon à minimiser la quantité chaleur fatale. Les données des procédés sont combinées afin de réduire le procédé industriel à l'étude simple des flux chauds et froids en fournissant trois paramètres cruciaux issus des capteurs :

- La température d'entrée notée T_e
- La température de sortie notée T_s
- débit noté C_p

Ces données ainsi obtenues nous fournissent des courbes composites composées d'une part de l'ensemble des flux chauds et d'autre part l'ensemble des flux froids. Le point où l'approche en température entre les courbes composées « chaudes » et « froides » est la plus faible sera la température de pincement. En localisant ce point et en commençant la conception à cet endroit, les consommations énergétiques minimum peuvent être atteintes en mettant en place des échangeurs pour récupérer la chaleur entre les courants chauds et les courants froids.
Un exemple de courbe composite est représenté sur la figure Fig. 7 par ces flux chauds en rouge et ces flux froids en bleu.

2. flux chaud : flux nécessitant un refroidissement
3. flux froid : flux nécessitant un chauffage

Désignation	T in (°C)	T out (°C)	Débit
Refroidissement du compresseur d'air	65	40	4.2
Refroidissement du four de fusion	65	40	4.2
Production d'eau chaude	15	60	4.2
Chauffage des bâtiments	45	55	4.2

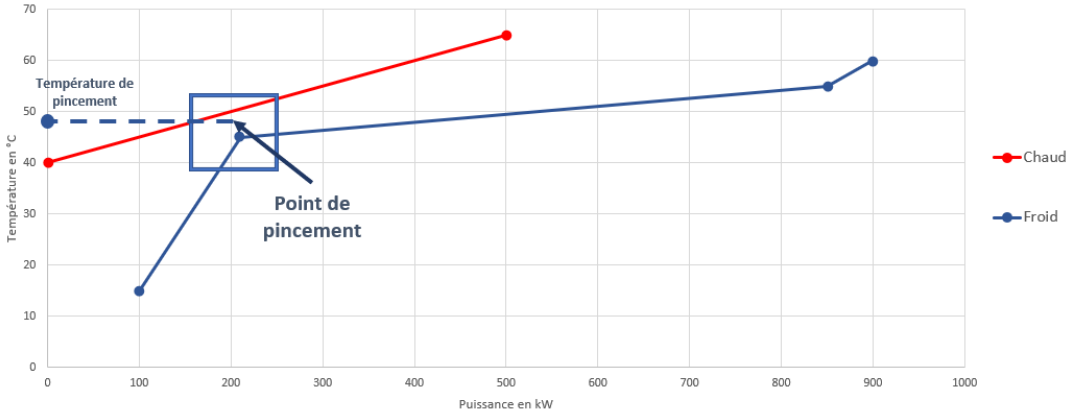


FIGURE 7 – Exemple de courbe composite

Quelques règles doivent être cependant respectées afin de créer un réseau :

- Diviser le problème en deux définissant le dessus et le dessous du pincement
- Pas de transfert à travers le pincement
- Pas d'utilités chaudes en dessous du pincement
- Pas d'utilités froides au-dessus du pincement

Les principaux avantages de cette approche sont :

- C'est une méthode graphique et visuelle afin d'avoir une approche physique du phénomène.
- L'ensemble des procédés sont pris en compte dans sa globalité.
- Elle ne nécessite pas la connaissance préalable du réseau d'échange qui n'est défini qu'ultérieurement.
- Capable de réduire les investissements et les coûts d'exploitation.

Nous cherchons à obtenir des réseaux d'échangeur optimaux afin d'obtenir les meilleurs rendements possibles.

2.5.3 Application de la méthode de pincement : Segmentation

La segmentation de la série temporelle est utilisée pour extraire des segments dans lesquelles le signal est considéré comme constant et pour identifier les points de ruptures de cette dernière. Le principe de cette section est d'étudier l'approche de la segmentation multivariée qui regroupe les informations de plusieurs capteurs du système de telle sorte à qu'elle ajoute des informations supplémentaires comparées à une analyse et une segmentation isolée de chacun des flux qui risque de générer un nombre de segments trop élevé.

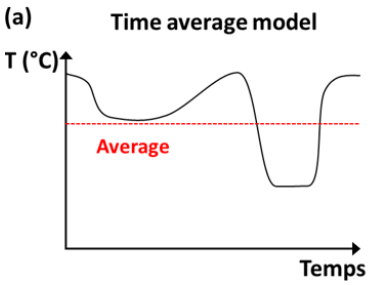
Afin d'appliquer la méthode de pincement, il est nécessaire de fournir la ou les séries temporelles contenant toutes les évolutions des flux dans le temps et intervenant dans le procédé afin de pouvoir mettre en place un réseau d'échangeur. L'approche de l'application de la méthode de pincement se base alors sur trois différentes techniques possibles pour fournir ces derniers [21] :

- Time average model

- Time slice model (TSM)
- adaptative segmentation model (ASM)

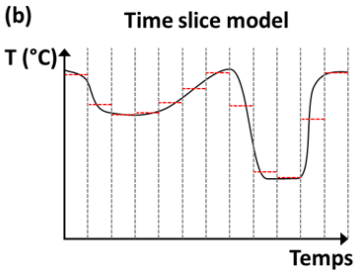
Ces différentes méthodes de segmentation fourniront un ou plusieurs intervalles de temps à laquelle seront prises les valeurs moyennes. Nous appliquerons alors sur **chacun** de ses intervalles de temps la méthode de pincement afin d'obtenir notre réseau d'échangeur. Il est cependant très important de savoir que le réseau final est la **somme** des réseaux optimisés obtenus sur chaque intervalle de temps, ce qui signifie que plus nous appliquons la méthode de pincement et plus notre réseau d'échangeur risque de devenir complexe.

Time average model : .



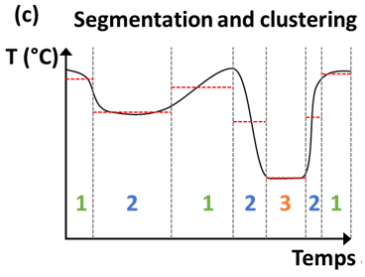
Le principe pour appliquer la méthode de pincement est de simplement considérer les valeurs moyennes des paramètres sur toute la longueur de la série comme le montre la figure suivante : temporelle.
L'avantage : Elle est très simple d'utilisation et qu'elle est peu coûteuse en termes de temps d'application.
L'inconvénient : Elle n'est pas appropriée pour les procédés plus complexes possédant plusieurs variations temporelles, ce qui n'apporte pas de solutions optimales et précises.

Time slice model : .



Au lieu de prendre la valeur moyenne sur toute la durée de la série temporelle, l'approche dynamique consiste à segmenter le flux en plusieurs intervalles de temps réguliers dans lesquels on appliquera la méthode de pincement sur chacun de ces intervalles en prenant cette fois-ci la moyenne de la valeur des paramètres sur ce dernier.
L'avantage :Méthode très précise pour quantifier la quantité de chaleur récupérable, permet d'identifier les échanges directs entre les flux lorsqu'ils existent, ainsi un réseau d'échangeurs est créé sur chaque intervalle de temps. Le réseau final est la somme des réseaux optimisés sur chaque intervalle de temps.
L'inconvénient : Lorsque la segmentation est trop fine, le temps de calcul est très long et fournit une solution trop complexe et trop coûteuse.
Il est alors intéressant d'appliquer la segmentation automatique afin d'avoir des solutions plus abordables tout en conservant la précision.

Segmentation automatique : .



Le principe est d'adapter les valeurs moyennes dans des intervalles de temps où les variations des flux semblent constant.
L'avantage : Fournit des résultats très satisfaisant avec un coût en temps de calcul et une complexité du réseau d'échangeur bien inférieur que celui du `Time slice model`.
L'inconvénient : On obtient des résultats moins optimaux que ceux du Time slice model et la complexité du réseau d'échangeur peut rester élevée.

2.5.4 Bottom-up segmentation

Afin d'obtenir notre segmentation automatique, nous utilisons l'algorithme **Bottom-up segmentation** [16] qui consiste dans un premier temps à diviser notre série temporelle en plusieurs petits segments pour ensuite fusionner les segments voisins les plus homogènes à chaque itération. Cette notion d'homogénéité va dépendre d'une fonction de coût que l'on nommera `fusion()`. Ainsi, l'idée sera de fusionner deux segments voisins qui seront les moins "coûteuses" à fusionner. La méthode est illustrée par la figure Fig. 8 suivante :

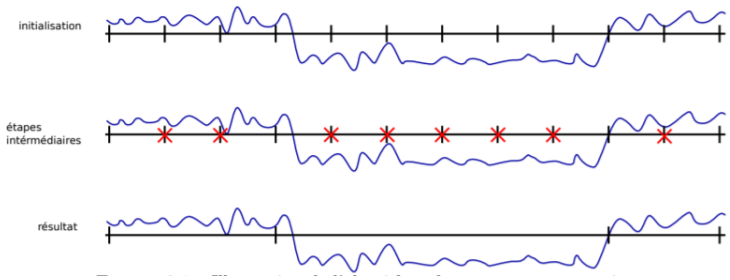


FIGURE 8 – Illustration de l'algorithme Bottom-up segmentation.

La partie segmentation n'étant pas mon étude principale, elle ne sera pas expliquée en détail dans mon rapport.

2.5.5 Clustering et RDSC

Le clustering aura pour objectif de regrouper et de classifier les segments les plus similaires dans notre série temporelle. En rajoutant une étape de clustering sur notre segmentation multivariée où on appliquait la méthode de pincement sur chacun des segments obtenus, nous appliquerons le modèle thermique seulement sur les clusters de segments similaires afin de réduire le nombre d'intervalles de temps. En effet, puisque le nombre d'intervalles de temps a un impact sur la complexité du réseau final, il est judicieux de réduire ce dernier. Par conséquent, afin de regrouper les segments, nous avons opté pour deux méthodes dont l'une est la poursuite du travail effectué par les stagiaires précédents basé sur l'utilisation de **facteurs de similarité** en réduisant la dimension de l'espace des variables de chaque segment [12, 1, 23, 25]. Deux segments sont projetés dans un sous-espace vectoriel et un critère de similarité est appliqué pour déterminer la similarité entre ces deux segments à travers une combinaison convexe de deux fonctions de coût, l'une se focalisant sur l'orientation spatiale et l'autre sur la localisation spatiale. La deuxième méthode **TICC** est bien plus récente et bien plus spécifique à la série temporelle avec utilisation de capteur et sera détaillée dans la section 2.5.7. Considérons alors deux segments X_1 et X_2 de dimension respectivement (n, t_1) et (n, t_2). La première méthode présentée ci-dessous sera appelée RDSC pour Reducing Dimensions of the variable Space for Clustering.

Premier facteur de similarité .

Nous définissons tout d'abord notre espace vectoriel dont les vecteurs de base seront :

- ☐ soit les vecteurs propres obtenus lors de la décomposition en valeurs singulières (Annexe : 5.2).
- ☐ soit les vecteurs propres obtenus lors de l'analyse en composantes principales (Annexe : 5.2).

Nous prenons ensuite les $n-1$ premiers vecteurs de base de chacun des segments X_1 et X_2 qui seront donc des sous-espaces vectoriels. Ainsi, notre notion de similarité entre nos deux segments sera la similarité

de nos deux sous-espaces vectoriels (que l'on notera S_{SEV}) calculé de la manière suivante :

$$S_{SEV} = \frac{\sum_{i=1}^{n-1}\sum_{j=1}^{n-1}(\lambda_{i,1}\lambda_{j,2})cos^2(\theta_{ij})}{\sum_{i=1}^{n-1}\lambda_{i,1}\lambda_{j,2}} \tag{1}$$

où θ_{ij} est l'angle entre le i-ième vecteur de X_1 et le j-ième vecteur de X_2

Second facteur de similarité .

Il est possible que deux segments aient la même orientation spatiale, mais qui pour autant sont différents en termes de localisation dans l'espace. Pour palier ce problème, un second facteur de similarité noté S_{dist} est intégré comme suit :
Notons F la fonction de répartition de ϕ tel que :

$$F(\phi) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\phi} e^{-z^2/2}dz$$

avec $\phi = \sqrt{(\bar{x}_2 - \bar{x}_1)\Sigma_1^{*-1}(\bar{x}_2 - \bar{x}_1)^T}$
$\bar{x}_i$ représentent les moyennes des X_i et Σ_1^{*-1} est le pseudo inverse de la matrice de covariance de X_1.
Nous avons alors :

$$S_{dist} = 2 \times [1 - F(\phi)] \tag{2}$$

Nous combinons alors ces deux facteurs de similarité notés SF :

$$SF = \alpha S_{SEV} + (1 - \alpha)S_{dist} \tag{3}$$

où $\alpha \in [0, 1]$.

2.5.6 Classification hiérarchique :

Afin de regrouper les segments similaires dans un cluster, nous utilisons la classification hiérarchique agglomérative dont le principal intérêt est qu'elle ne nécessite pas en entrée le nombre de cluster et qu'elle regroupe petit à petit les objets qui sont similaires jusqu'à avoir un seul groupe.
Principe :
Nous définissons la dissimilarité entre deux segments X_i et X_j suivante :

$$d_{ij} = 1 - SF_{ij} \tag{4}$$

avec SF_{ij} définie par (3).
Le principe sera de chercher tout d'abord les clusters avec la dissimilarité la plus faible (les plus similaires). Chaque cluster est composé d'un seul segment à la première itération, puis au fur et à mesure du regroupement, le nombre de cluster s'amoindrit en regroupant un ou plusieurs segments. Lorsque deux ou plusieurs segments sont regroupés dans un cluster, on doit redéfinir la dissimilarité pour ce cluster avec les autres clusters. Ceci se faisant par l'étape ($\star$) suivante :
Soit U, V deux clusters similaires regroupés dans un nouveau cluster (UV) et, soit W un autre cluster, la dissimilarité entre le cluster (UV) et W se calcule à l'aide du choix des distances [28] suivantes :

- ☐ **Single linkage** : Elle représente simplement la distance minimum entre les segments du cluster (UV) et les segments du cluster W.

$$d_{(UV)W} = min(d_{UW}, d_{VW})$$

- ☐ **complete linkage** : Même principe mais cette fois-ci c'est la distance maximale entre les segments du cluster (UV) et les segments du cluster W.

$$d_{(UV)W} = max(d_{UW}, d_{VW})$$

☐ **average linkage** : Elle représente la distance moyenne entre les segments du cluster (UV) et les segments du cluster W.

$$d_{(UV)W} = \frac{\sum_i \sum_j (d_{ij})}{N_{(UV)} N_W}$$

où d_{ij} est la distance entre le i-ième segment dans le cluster (UV) et le j-ième segment dans le cluster W et $N_{(UV)}$ et N_W représente respectivement le nombre de segments dans le cluster de (UV) (resp. W).

☐ **barycentre distance** : Elle représente la distance entre le barycentre des segments du cluster (UV) et le barycentre des segments du cluster W.

$$d_{(UV)W} = d(\mu_1, \mu_2)$$

où μ_1, μ_2 sont les centres de gravité de (UV) et W.

☐ **Ward distance** : Elle représente la distance entre les barycentres des deux clusters au carré, pondérée par les effectifs des deux clusters.

$$d_{(UV)W} = \frac{d(\mu_1, \mu_2)^2}{1/N_{(UV)} + 1/N_{(W)}}$$

☐ **Beta-flexible distance** Elle représente la distance moyenne pondérée par $(1 - \beta)$ entre les segments du cluster (UV) et W et la distance moyenne pondérée par β entre les segments du même cluster (UV) et W.

$$d_{(UV)W} = (1-\beta)\frac{\sum_i \sum_j (d_{ij})}{N_{(UV)} N_W} + \beta \frac{\sum_k \sum_l (d_{kl})}{N_{(UV)}} + \beta \frac{\sum_m \sum_n (d_{mn})}{N_W}$$

où d_{kl} respectivement d_{mn} est la distance entre k-ième (resp m-ième) segment et le l-ième (resp n-ième) segment du cluster (UV) (resp W) et $\beta \in [0, 1]$

L'algorithme 2 décrit l'implémentation de notre classification hiérarchique agglomérative.

Algorithme 2 : Clustering agglomératif

1 On construit la matrice $N \times N$ de dissimilarité symétrique $D = \{d_{ij}\}$;
2 **pour** $i \leftarrow 1$ **à** $N - 1$ **faire**
3 On cherche les clusters les plus similaires (qui ont le minimum de dissimilarité). Posons U et V les clusters plus similaires, et la dissimilarité est noté d_{UV};
4 On rassemble U et V en un cluster (UV). On met à jour la matrice D :
  • Supprimer les colonnes et lignes correspondants à U et V
  $(\star)$ Ajouter une ligne et colonne pour (UV) et recalculer la dissimilarité entre (UV) et les clusters restants ;
5 **fin**

2.5.7 Toeplitz Inverse Covariance-based Clustering

Contrairement aux principes de débuter par une segmentation de notre série temporelle puis qui nous conduit au clustering, une méthode alternative propose d'alterner simultanément la segmentation et le clustering de nos séries temporelles multivariées à travers l'utilisation de "Markov random fields" (MRF) [10]. La raison d'utiliser un réseau de Markov est qu'il permet généralement de mieux mettre en évidence les relations entre les variables plutôt qu'avec une simple méthode de corrélation. Cette méthode alternative se base sur le **Toeplitz Inverse Covariance-based Clustering (TICC)**.

Considérons notre série temporelle multivariée avec plusieurs variables représentant les capteurs dans les procédés. La figure suivante Fig. 9 illustre un exemple d'une série temporelle multivariée qui sera notre objet d'étude.

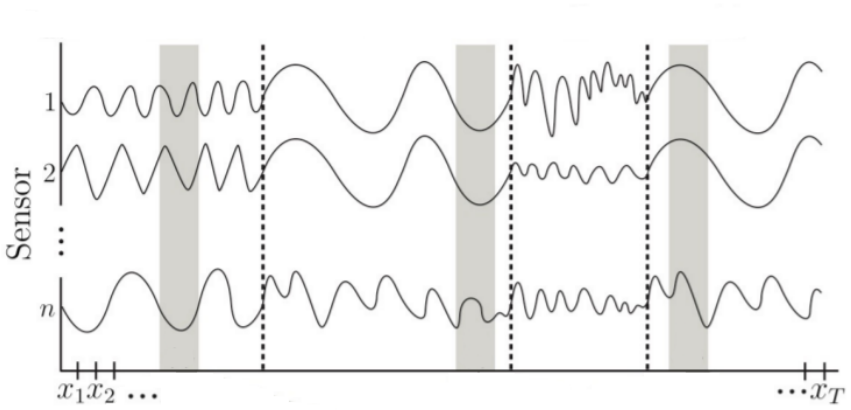


FIGURE 9 – Exemple de série temporelle multivariée

Considérons comme sur la figure Fig. 9 une série temporelle avec T observations séquentielles $x_1, x_2, ..., x_T$. L'objectif est donc de regrouper ces T observations dans K clusters. Pour parvenir à tel résultat, nous intégrons une fenêtre de longueur $w << T$ afin que chaque observation ne soit pas isolée et qui prend en compte ces prédécesseurs. cela passe par définir une nouvelle variable X_T qui concatène en ligne les observations $x_{t-w+1}, ..., x_t$ dans un vecteur de dimension nw. Nous pouvons ainsi observer la sous-séquence $x_{t-w+1}, ..., x_t$ en observant X_T comme l'illustre la figure suivante Fig. 10 :

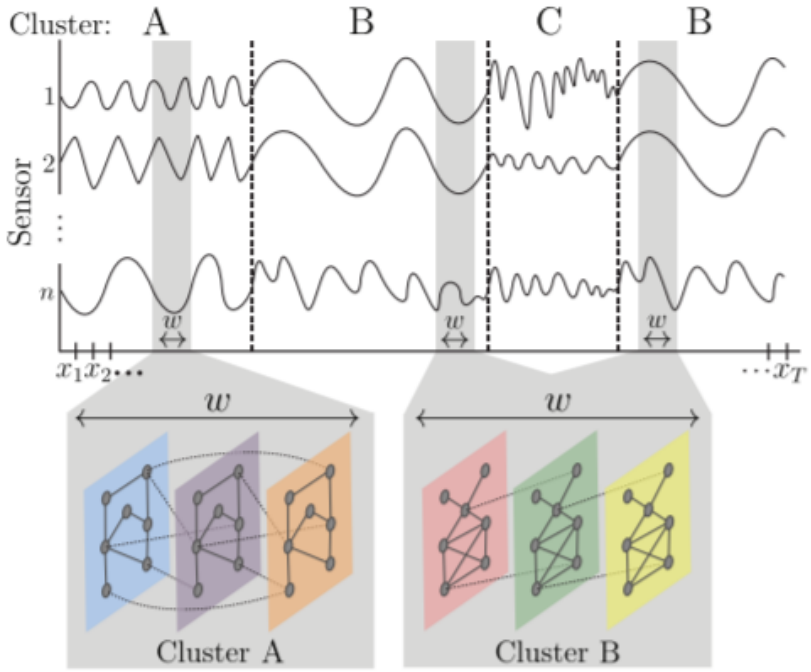


FIGURE 10 – La fenêtre w dans la série temporelle

Ainsi chaque cluster sera défini comme un réseau de dépendance prenant en compte l'interdépendance de nos n variables d'une observation x_t et la dépendance de ces voisins. Chaque réseau d'un cluster peut être représenté par un graphe non orienté multicouche où les sommets représentent nos variables et les arêtes représentant leur dépendance comme illustré par les figures Fig. 11, 12.

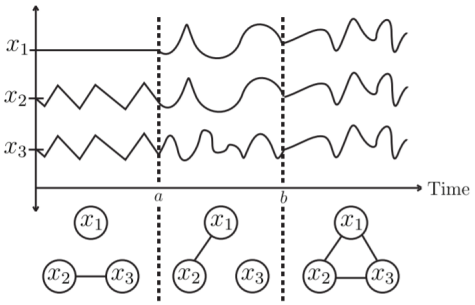


FIGURE 11 – MRF structure entre chaque observation x_t

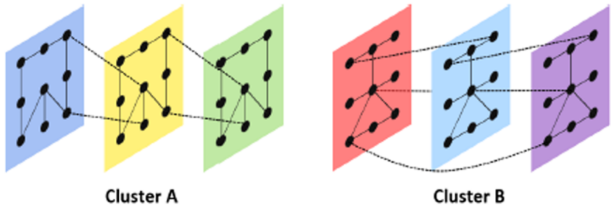


FIGURE 12 – MRF structure entre deux clusters

Nous représentons ces clusters par une covariance inverse gaussienne $\theta_i \in \mathbf{R}^{nw \times nw}$ qui mettra alors en évidence l'indépendance conditionnelle des structures entre les variables comme expliqué ci-dessus.

L'approche de l'algorithme `TICC` est de résoudre le problème d'affectation des observations x_t dans un cluster qui détermine l'ensemble $P = \{P_1, ..., P_K\}$ où $P_i \subset \{1, 2, .., T\}$. Par ailleurs, l'algorithme doit mettre à jour les paramètres des clusters $\theta = \{\theta_1, ..., \theta_K\}$ calculé à l'aide de notre ensemble P. Notre problème d'optimisation peut être formulé comme :

$$\underset{(\theta,P)}{\operatorname{argmin}} \sum_{i=1}^{K} = \left[||\lambda \circ \theta_i||_1 + \sum_{X_t \in P_i} \left(- ll(X_t, \theta_i) + \beta \mathbb{1}_{X_{t-1} \notin P_i} \right) \right] \tag{5}$$

où τ représente l'ensemble les matrices de dimension $nw \times nw$ des blocs symétriques Toeplitz auxquelles on ajoute des contraintes sur la construction des θ_i (Annexe [5.3]).

Termes :

- ☐ La première expression $||\lambda \circ \Theta_i||_1$ représente la contrainte de parcimonie basée sur le produit d'Hadamard de la matrice de covariance inverse paramétré avec un paramètre de régularisation $\lambda \in \mathbb{R}^{nw \times nw}$. Elle permet de jouer sur la forme creuse de la matrice Θ_i et donc de modifier le nombre d'arêtes illustré sur les figures Fig. 11, 12.
- ☐ La deuxième expression $ll(X_t, \Theta_i)$ permet de déterminer la log-vraisemblance que notre observation X_T soit dans le cluster i.
- ☐ La dernière expression représente la cohérence temporelle qui ajoute une pénalité β afin que deux observations consécutives soient assignées au même cluster ou non et donc dans un même segment.

Un problème est que le couple de nos ensembles de paramètres (Θ,P) rend notre problème très non convexe, ce qui implique que nous aurons du mal à converger vers une solution globale. Pour palier ce problème, nous utilisons une variante grâce à l'algorithme espérance-maximisation (ME) (expectation-maximization (EM) en anglais) qui permet d'alterner entre regrouper des points dans un cluster et mettre à jour nos paramètres de cluster afin de résoudre ce problème d'optimisation.

Regroupement des points dans un cluster (Espérance) : .

Comme dans l'étape "espérance" d'un algorithme ME classique, l'algorithme TICC commence à regrouper des observations dans un cluster donné. Nous regroupons les observations dans des clusters en fixant la valeur

de Θ et nous résolvons le problème d'optimisation suivant pour $P = \{P_1, ..., P_K\}$:

$$\min_{P} \sum_{i=1}^{K} \sum_{X_t \in P_i} -ll(X_t, \Theta_i) + \beta \mathbb{1}\{X_{t-1} \notin P_i\} \tag{6}$$

Ainsi chaque sous-séquence X_t est regroupée dans un cluster K en fonction du maximum de vraisemblance tout en ayant une certaine cohérence temporelle (on veut minimiser les points de ruptures). Le problème 6 est résolu en utilisant l'algorithme de Viterbi. Cette méthode est équivalente à chercher le cout du chemin minimum entre le temps 1 à T comme présenté sur la figure Fig. 13 suivante :

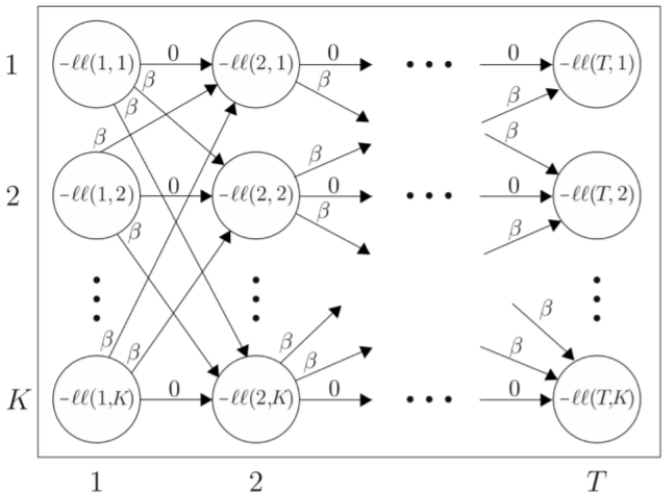


FIGURE 13 – Coût du chemin minimum de 1 à T

Mettre à jour Θ (Maximisation) : .

L'étape de maximisation de notre algorithme `ME` consiste à mettre à jour la matrice de covariance inverse des clusters une fois que l'ensemble P à l'étape `E` soit construit. Pour cette étape, c'est l'ensemble P qui est fixé et dont on ajuste les Θ_i en conséquence. Comme à l'étape (E) nous allons résoudre un sous-problème plus simplifié du problème (5) en résolvant l'expression suivante :

$$\underset{\Theta \in \tau}{\operatorname{argmin}} \sum_{i=1}^{K} \left[||\lambda \circ \Theta_i||_1 + \textstyle\sum_{X_t \in P_i} -ll(X_t, \Theta_i) \right] \tag{7}$$

Une propriété pertinente de ce problème d'optimisation est que le problème de mise à jour pour chaque cluster peut être calculé indépendamment, car il n'y a pas de dépendance entre les termes précédents. Par conséquent, toutes les mises à jour peuvent être faites en parallèle plutôt que séquentiellement [?]. Afin d'apporter le problème global dans une forme plus facile à manipuler, nous devons réarranger la log-vraisemblance négative qui peut être exprimée comme suit :

$$\sum_{X_t \in P_i} -ll(X_t, \Theta_i) = -|P_i|(log(det\Theta_i)) + tr(S_i\Theta_i)) + C \tag{8}$$

où $|P_i|$ est le nombre d'observations X_t regroupé dans le cluster i, S_i est la matrice de covariance empirique de ces points et C une constante indépendante de Θ_i. Le problème (7) peut ainsi se simplifier par l'expression suivante :

$$\min_{\Theta_i \in \tau} -log(det(\Theta_i)) + tr(S_i\Theta_i) + \frac{1}{|P_i|}||\lambda \circ \Theta_i||_1 \tag{9}$$

Afin de résoudre notre problème (9), nous allons utiliser l'algorithme `ADMM` (Alternating Direction Method of Multipliers) dont de nombreux problèmes convexes peuvent être résolus de manière très efficace et de manière séquentielle en utilisant cet algorithme. Cependant, afin que cette approche soit applicable, nous avons besoin encore de reformuler notre problème (9) afin de respecter la forme requise pour l'algorithme `ADMM`.
Nous introduisons dès lors une variable Z afin que notre fonction objective soit divisée en deux fonctions objectives indépendantes satisfaisant la forme requise de l'algorithme `ADMM` et nous voulons résoudre le problème suivant :

$$\min_{\Theta=Z, Z \in \tau} -log(det(\Theta)) + tr(S\Theta) + ||\lambda \circ Z||_1 \tag{10}$$

Les deux étapes de regroupement des points dans un cluster (E) et la mise à jour des paramètres du cluster (M) se réitère jusqu'à ce que le regroupement des points deviennent stationnaires. La méthode `TICC` peut se résumer par l'algorithme 1 suivant :

Algorithm 1 TICC (high-level)

Require: Cluster paramaters Θ, cluster assigments P
  while Non-stationary **do**
    E-step : Assign points to cluster $\rightarrow P$
    M-step : Update cluster paramaters $\rightarrow \Theta$
  end while
  return Θ, P

2.6 Expérimentation

À travers cette étude, de nombreux algorithmes ont été explorés dans lesquels chacun d'eux possèdent leurs propres particularités ou paramètres, ce qui rendent la paramétrisation des modèles difficiles. Chaque décision prise en vue d'obtenir les performances optimales de clustering est basée sur des critères mathématiques de sélection pour avoir une procédure automatique de décision. Cependant, le clustering de notre série temporelle dépend grandement de l'interprétation via l'utilisation d'un modèle de référence. Ainsi, selon le modèle de référence choisi, certains résultats peuvent être différents. De plus, puisque le choix de la segmentation affecte également l'efficacité des modèles $RDSC$, il est judicieux d'isoler les intervalles de temps contenant de grandes variations avec une détection des points de ruptures afin d'améliorer les performances des modèles $RDSC$. Les paragraphes suivants présentent les différentes expérimentations faites pour atteindre nos objectifs.

2.6.1 Segmentation :

Les étapes de la segmentation débutent par diviser notre série temporelle en plusieurs petits segments. Dès lors, il nécessite un nombre de segments initial que l'on doit fixer. Pour faciliter la comparaison des résultats obtenus, ce nombre initial a été fixé à 160 000 et le nombre final à 20. Puis, comme expliqué dans l'article Spiegel et al [27], il est essentiel que nos données soient mises à la même échelle, donc normalisées. L'indépendance entre la segmentation et l'algorithme TICC est un très gros atout pour les performances de clustering de ce dernier. En effet, la sélection d'une fonction de coût, des distances pour la classification hiérarchique agglomérative étant difficile, avoir la possibilité de dépendre seulement de ses propres paramètres est un très grand avantage et rend ce modèle plus efficace, flexible et robuste.

2.6.2 TICC :

L'algorithme TICC possède un total de 4 paramètres afin d'obtenir un modèle : le nombre de cluster, la fenêtre w et les coefficients α et β. Bien que sélectionner les meilleurs paramètres ne semble pas évident, le critère d'information bayésien (BIC) détaillé dans la partie Annexe 5.4 tente de résoudre ce problème et est très efficace pour les sélectionner de manière optimale. C'est un critère de sélection parmi un ensemble fini de modèle qui, lors de la sélection entre plusieurs modèles, ceux dont les valeurs de BIC sont plus faibles sont généralement préférés. Nous appliquons cette méthode pour trouver quel modèle semble le plus performant sur chacune des valeurs de cluster fixées pour différents paramètres. Cependant, une valeur de BIC très faible pour une certaine valeur de cluster ne signifie pas que la meilleure valeur de cluster est celle-ci. Ensuite, pour éviter de générer un nombre conséquent de modèles TICC avec ces valeurs de BIC correspondant afin de trouver la valeur minimale de BIC pour un nombre de cluster fixé, nous avons intégré à notre processus un algorithme méta-heuristique (NSGA-II) pour converger rapidement vers une valeur de BIC optimale sans avoir la nécessité d'explorer un nombre trop important de paramètre. Malheureusement, ces modèles ont certains défauts tels qu'un temps de calcul long dû à la fenêtre w lorsque celle-ci est grande. Par conséquent, nous avons décidé de fixer une plage de valeur possible pour les paramètres afin d'une part réduire les temps de calcul et d'autre part améliorer les performances de convergence. Le premier terme α sera fixé comme suit : $\alpha \in [0, 10]$ puisqu'il n'infère peu dans les performances du clustering. Le nombre de cluster quant à lui sera défini entre 3 et 9, une valeur plus élevée ne sera pas productive car elle ne représentera plus réellement les phases du procédé industriel que l'on peut observer. Ensuite, β est un coefficient important du modèle car il force la cohérence temporelle afin que les points voisins soient assignés au même cluster. Il est donc intéressant de choisir une large palette de valeur possible de cette dernière pour pouvoir étudier le phénomène. Pour cette raison $\beta \in [0, 500]$. Finalement, puisque la fenêtre w augmente considérablement le temps de calcul lorsque celle-ci est importante, nous l'avons fixé à des valeurs relativement faibles avec $w \in [\![4;\ 7]\!]$

2.6.3 RDSC :

Les modèles $RDSC$ dépendent de la méthode de segmentation avec sa fonction de coût associée. Nous avons utilisé la fonction de coût L_2 dans cette étude en raison de sa capacité à trouver les points de rupture efficacement, comme expliqué ultérieurement dans la section. De plus, une fois la segmentation faite, le modèle RDSC est plutôt simple à paramétrer et à exécuter car il nécessite seulement 2 paramètres : $\alpha \in [0, 1]$ (pour donner de l'importance à $S_{SEV} ou S_{dist}$) et la distance à utiliser lors de la classification hiérarchique agglorémative. Pour des raisons de simplicité et de cohérence (être flexible et robuste pour toutes formes de série temporelle), nous avons utilisé la "average distance" pour cette classification. Puis, bien que le BIC ne soit pas applicable dans ce contexte, nous avons opté pour le $F1$-score (détaillé dans Annexe 5.5) qui est l'une des techniques de mesure les plus populaires pour évaluer l'efficacité d'un classificateur de manière à pouvoir sélectionner α. Cependant, cette méthode nécessite plusieurs clustering de série temporelle de références obtenus par des annotateurs pour pouvoir comparer entre ce que notre modèle décide de regrouper et ce que le modèle de référence a considéré comme regroupement des segments. Dans notre cas $\alpha = 1$ semblait être recommandé par notre $F1$-score. À noter qu'on aurait également pu utiliser le $F1$-score pour le modèle TICC pour sélectionner le meilleur modèle, cependant, la méthode BIC ne nécessite pas un modèle de clustering de référence, ce qui est préférable puisque l'obtention de cette dernière peut être fastidieuse.

2.6.4 Réseau d'échangeur :

Une fois notre segmentation construite et notre cluster obtenu lors de toutes les étapes précédemment détaillées, nous récupérons toutes ces données afin d'appliquer la méthode d'analyse par pincement et d'établir notre réseau d'échangeur optimal. Ceci se faisant à l'aide d'une boite-noire du procédé MidRex qui permet le dimensionnement des réseaux d'échangeurs, d'utilités et tuyauteries tout en prenant en compte divers paramètres comme l'emplacement des flux étudiés dans le site industriel, le choix du type d'échangeurs, le positionnement des pompes etc...

Voici les principales caractéristiques de la boite noire :

- ☐ nécessite en entrée la position des flux étudiés dans le site industriel et les valeurs moyennes des paramètres des flux sur chaque intervalle de temps crée par notre clustering.
- ☐ va positionner de manière optimale les échangeurs, utilités, tuyauterie.
- ☐ retourne comme sortie différentes valeurs énergétiques et financières : le MER, le $\%MER$, le CAPEX (coût d'investissement) et l'OPEX (coût d'exploitation), les économies en CO_2, l'énergie échangée etc ...

Nous avons des critères énergétiques et financiers détaillés comme suit :

- **MER** : Minimum d'énergie requis qui permet de déterminer en kW la quantité d'énergie externe qu'on devra apporter au procédé afin de satisfaire ces besoins.
- **% MER** : l'écart relatif entre le `MER` et l'énergie requise (noté ER par les utilités) :

$$\%MER = \frac{100 \times (ER - MER)}{MER}$$

- **CAPEX** : le coût d'investissement de la mise en place total du réseau d'échangeur optimal.
- **OPEX** : la dépense d'exploitation représentée par le nouveau coût annuel des énergies consommées lors des procédés.
- **L'énergie échangée** : L'énergie échangée par le réseau d'échangeur en kW.

2.7 Résultats

2.7.1 Segmentation :

La Fig 14 montre les résultats des algorithmes de segmentation effectués par les précédents stagiaires.

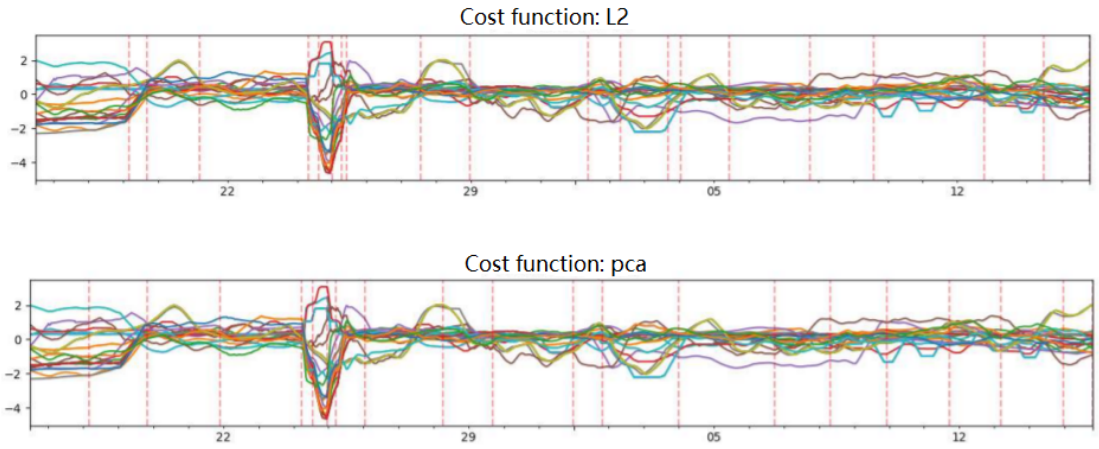


FIGURE 14 – Résultats de la segmentation sur l'ensemble des flux de notre procédé

La méthode basée sur la fonction de coût L_2 a semblé efficace pour la détection des points de ruptures. Il réussit à trouver l'ensemble des changements de variations importants car il est plus sensible aux changements des valeurs moyennes. Cela s'observe sur la période du 1 février au 6 février et du 27 janvier au 29 janvier où différentes variations ont été captées et détectées par notre algorithme. Comme l'objectif est d'estimer la quantité de chaleur récupérable, il est souhaité d'isoler les intervalles de temps contentant de grandes variations, ce qui incite l'utilisation de la méthode L_2.

2.7.2 Clustering

Pour comparer le meilleur modèle TICC et le meilleur modèle RDSC, le choix s'est porté sur l'utilisation de la $F1$-score qui permet d'évaluer l'efficacité du clustering à travers des modèles de références. Puisqu'il est difficile de connaître la valeur idéale du nombre de cluster, nous avons décidé d'étudier le $F1$-score pour les valeurs suivantes 3,4 et 5 clusters.

Cependant, l'obtention d'un modèle de clustering de référence était nécessaire. Par conséquent, il a été demandé à mon tuteur Ufuk Halisdemir et à un stagiaire Matthias Foyer de proposer une solution de clustering sur 3,4 et 5 clusters, mais avec une seule contrainte : le premier segment devait se trouver dans un cluster unique. Ce choix de contrainte est dû par la nécessité d'appliquer la moyenne de manière cohérente de notre série temporelle pour l'intégration énergétique, or par crainte d'une interprétation différente lors du choix des clusters, il aurait été dommage pour l'application de notre modèle d'intégration énergétique de ne pas placer le premier segment dans un unique cluster dû à ses particularités et ses différences avec les autres segments. Néanmoins, notre F1-score possède certaines limites, d'une part notre clustering de référence aurait pu être obtenu soit grâce à une équipe ArcelorMittal afin d'obtenir une expertise métier, cependant l'équipe ArcelorMittal ne travaille plus sur ce projet, soit obtenir un plus grand nombre de modèles de clustering de référence à l'aide par exemple de différents employés ou stagiaires, mais qui, pour des raisons de simplicité n'a pas été mis en place. De plus, le choix des clusters reste souvent de l'ordre de l'interprétation et puisque nous voulions simplement un aperçu de la qualité des différents clustering, il nous aura pu bon de rester sur simplement 2 modèles de clustering de référence par cluster.

La figure Fig 15 montre les résultats du $F1$-score pour différentes valeurs de cluster. Les résultats du $F1$-score semble clairement indiquer que pour toutes les valeurs de cluster sélectionnées, le modèle TICC possède les meilleures performances par rapport au modèle RDSC, incitant ainsi à poursuivre les résultats avec ce modèle. En effet, la valeur du $F1$-score TICC est en moyenne 1,64 fois plus élevée que celle du modèle RDSC, ce qui est significatif. La figure Fig 16 illustre le clustering obtenu par les algorithmes RDSC (fig a) et TICC (fig b) pour une valeur de cluster fixée à 4 sur notre série temporelle. L'ensemble des modèles de référence et des résultats du clustering sont présentées dans les parties Annexe 5.7 et Annexe 5.8.

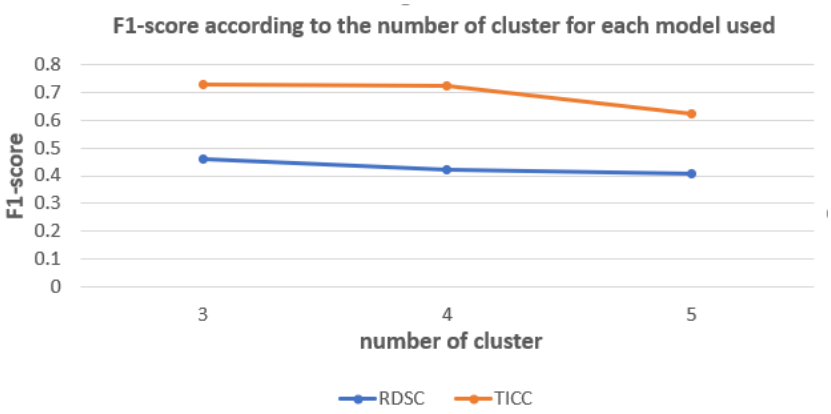
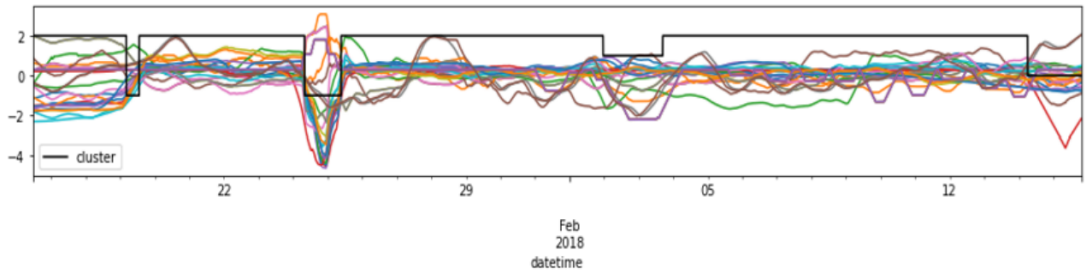
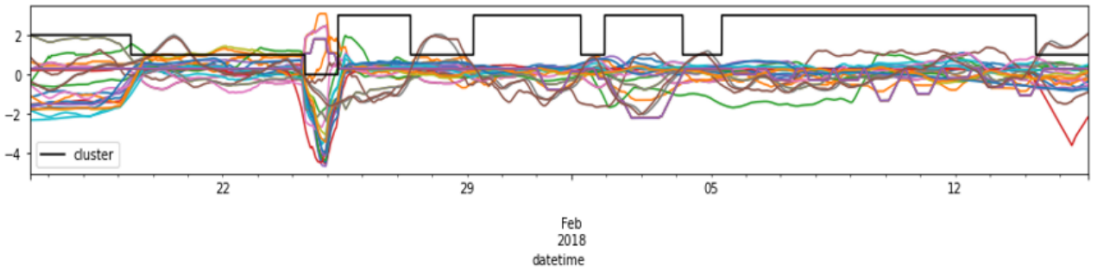


FIGURE 15 – Évolution du $F1$-score par rapport au nombre de cluster et du modèle choisis.



(a) Clustering du modèle RDSC pour 4 clusters



(b) Clustering du modèle TICC pour 4 clusters

FIGURE 16 – Clustering obtenu par les algorithmes RDSC (fig a) et TICC (fig b) pour 4 clusters

2.7.3 Réseau d'échangeur optimal

Cette partie présente tous les résultats générés provenant du modèle thermodynamique appelé boite-noire simulant les performances économiques et énergétiques du procédé MidRex. Plusieurs configurations possibles ont été considérées pour établir le réseau d'échangeurs telles que la méthode statique utilisant les valeurs moyennes de l'ensemble de la série temporelle, la méthode dynamique avec une segmentation adaptative qui divise notre série temporelle en plusieurs segments sur lesquels la méthode des pincements est appliquée sur chacun d'eux et enfin la méthode de clustering afin de regrouper les segments similaires pour pouvoir appliquer une seule fois sur chaque groupe la méthode de pincement.

Pour des raisons de compréhension, nous définissons "20" ou "20 clusters" la segmentation effectuée dans le contexte de la segmentation adaptative fournissant 20 segments et nous définissons "3,4,..9 clusters" les résultats obtenus avec les valeurs des clusters correspondant de notre série temporelle lors de l'étape de clustering. Enfin, "1" ou "1 cluster" se réfèrera à la méthode statique. Nous nous concentrons sur cette étude à l'évolution des performances énergétiques et économiques associées à chacune des valeurs de cluster. Ces performances économiques (resp. énérgétiques) sont le Capex, le nombre et le coût des échangeurs de chaleur (resp. la quantité d'énergie échangée, le pourcentage de MER et la puissance des utilités).

2.7.4 Etude du CAPEX, du nombre d'échangeur et du coût des échangeurs

Nous avions émis comme hypothèse que plus le nombre de clusters augmente et plus la valeur de **CAPEX ,du coût et du nombre d'échangeurs** augmentent en raison de l'augmentation de la complexité du réseau. Nous avons alors étudié l'évolution de ces derniers en fonction du nombre de clusters. Les résultats obtenus sont affichés sur les graphes Graphs. 17, 18,19 :

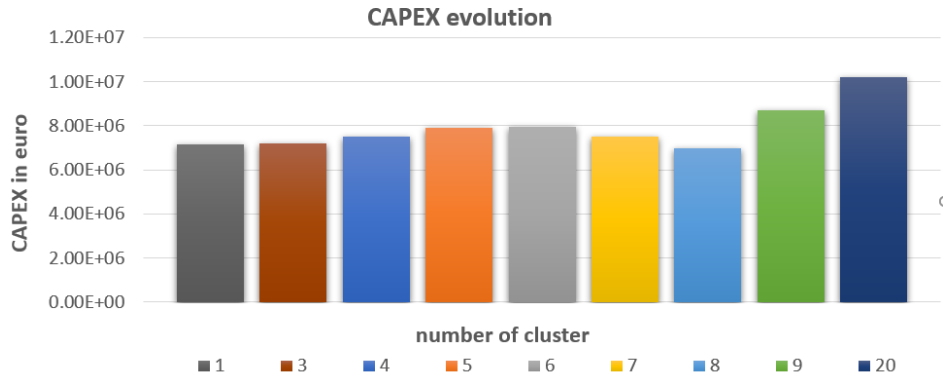


FIGURE 17 – Évolution du CAPEX

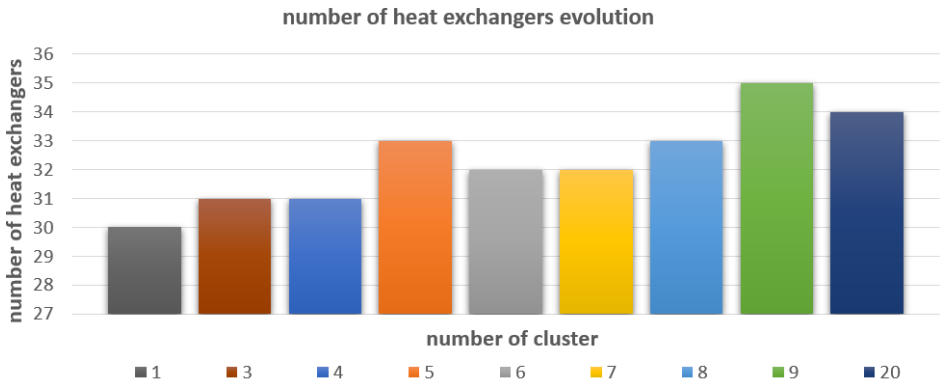


FIGURE 18 – Évolution du nombre d'échangeurs de chaleur

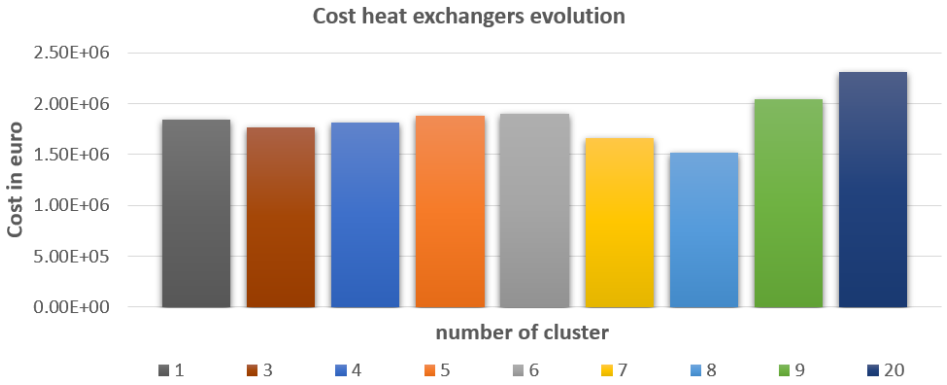


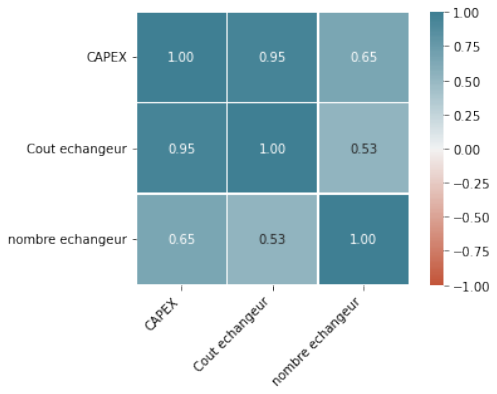
FIGURE 19 – Évolution du coût des échangeurs de chaleur

Nous observons globalement sur ces graphiques une augmentation de la valeur de CAPEX passant de 7 133 709 à 10 178 831 euros, du nombre (resp. du coût des échangeurs) passant de 30 à 35 (resp. environ $1.79E+06$ à $2.27E+06$) lorsque ce nombre de clusters augmente malgré une diminution de 14 % sur le CAPEX et 25 % sur le coût des échangeurs de chaleur lorsque ce nombre se situe entre 6 et 8. Cette diminution peut s'expliquer d'une part par une légère diminution du nombre d'échangeurs, mais également par une possibilité que la longueur des échangeurs, l'installation ou la complexité du réseau grâce à une proximité de certains échanges de flux entre les procédés soit moins importante. Cependant, nous observons

globalement une augmentation de ces valeurs, ce qui semble confirmer notre hypothèse. De plus, par rapport à la méthode statique, nous observons également que son coût engendré sur le CAPEX est inférieure à celle de la méthode de clustering ou de segmentation, ce qui semble également confirmer notre hypothèse initiale.

De plus, nous observons une très forte similarité sur les variations de ces derniers, dont plus spécifiquement entre l'évolution du CAPEX et l'évolution du coût des échangeurs, ce qui s'observe sur la matrice de covariance :

FIGURE 20 – Matrice de covariance du CAPEX, du nombre et du coût des échangeurs



Ces résultats sont dus en partie par une forte représentation du coût des échangeurs à hauteur en moyenne de **23%** du **CAPEX**, ce qui est conséquent.

2.7.5 Étude de l'énergie échangée, du pourcentage de MER et de la puissance des utilités

Sur les performances économiques, on a pu observer une tendance sur l'augmentation des coûts lorsque le nombre de clusters augmentait. Qu'en est-il sur les performances énergétiques ? Cette fois-ci, l'hypothèse était que plus le nombre de clusters ou segments augmente et plus **l'énergie échangée** augmente ou plus **le pourcentage de MER** diminue. Comme précédemment, nous affichons les résultats sur les graphiques Graphs. 21, 22 présentant l'évolution des performances énergétiques en fonction du nombre de clusters.

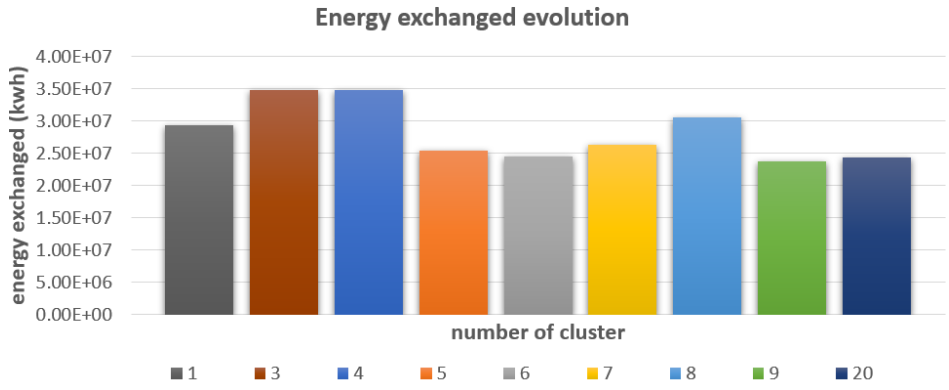


FIGURE 21

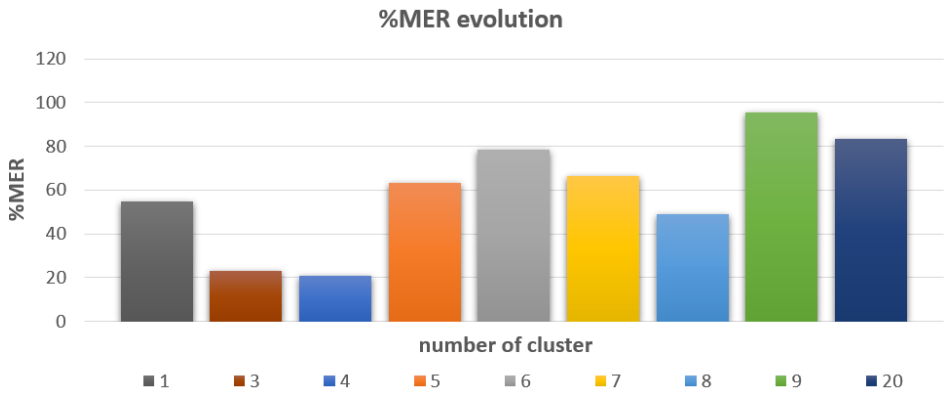


FIGURE 22

Une surprenante baisse des performances énergétiques est observée lors que le nombre de clusters augmente. En effet, on peut voir sur les Fig. 21,22 que l'énergie échangée est plus conséquente à hauteur de 43 % entre "3 et 20 clusters" et le pourcentage **MER** est bien plus basse [4] lorsque le nombre de clusters est faible. De plus, nous observons sur la figure Fig. 23 que les puissances d'utilités chaudes et froides croient légèrement lorsque le nombre de clusters augmente alors que le MER quant à lui diminue. La première possibilité est qu'en ayant un très faible nombre de clusters, nous moyennons de manière trop radicale notre série temporelle, ce qui induit une perte de précision sur le calcul du **MER** et de **l'énergie échangée** et que nous obtiendrions donc des valeurs de MER trop idéales pour les valeurs de trois et quatre clusters, ce qui expliquerait que les prochaines valeurs de MER sont enclins à bien mieux stagner à l'exception de huit clusters. Une autre possibilité serait que puisque le nombre d'échangeurs varie entre 31 et 35, ce qui est relativement proche, les 31 échangeurs sont simplement situées de manière plus optimale puisque les flux froids sont trop nombreux dans notre procédé et nécessite beaucoup d'énergie, ce que ne peut fournir les 5 flux chauds et donc rajouter plus d'échangeur aura l'aspect de brider les échanges des flux en redistribuant vers plusieurs utilités au lieu de se concentrer que sur certaine et ainsi d'éviter des pertes de chaleurs. Ceci s'observe sur la figure Fig. 23 puisque la puissance nécessaire des utilités chaudes (qui permet donc de chauffer les flux froids) est très importante par rapport à la puissance des utilités froides.

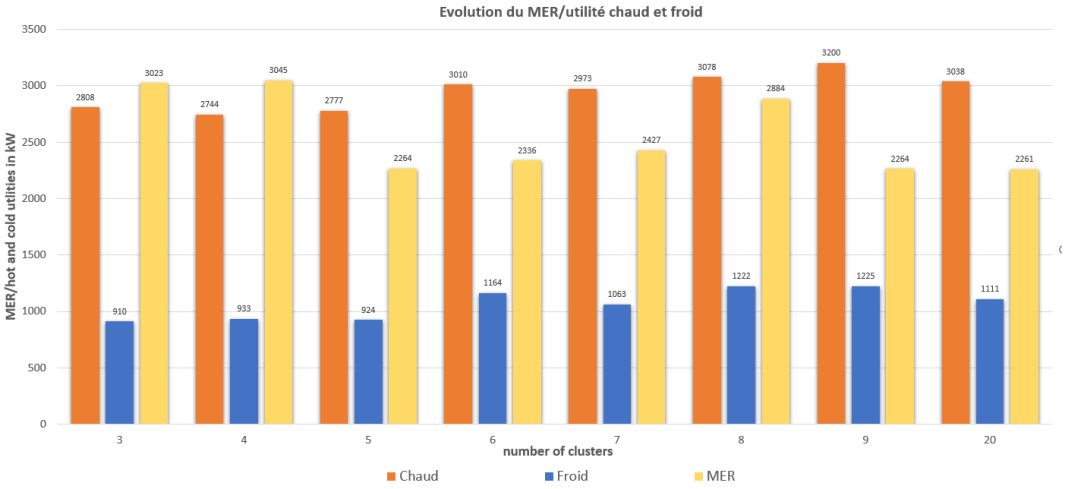


FIGURE 23 – Évolution du MER et des utilités chaude et froide (kw) dans le procédé MidRex.

Par rapport à la méthode statique, elle obtient de meilleures performances comparées à la méthode de

4. Rappel : un réseau d'échangeur idéal est un pourcentage de MER proche de 0

segmentation adaptative, mais est cependant moins performante que la méthode de clustering pour 3, 4 ou 8 clusters.

2.7.6 Front de Pareto

Nous souhaitons proposer aux clients différentes solutions viables économiquement et énergétiquement, ce qui correspond au CAPEX (ou du coût des échangeurs) et de certains critères de performance (énergie échangée, MER et économie totale). Cette approche aide le client dans leurs plans de développement avec des solutions selon son budget afin d'effectuer des économies tout en réduisant son empreinte carbone. Nous voulons ainsi minimiser le CAPEX et minimiser le pourcentage de MER ou maximiser l'énergie échangée et maximiser l'économie totale. Nous obtenons grâce à l'étude précédente et d'autre complémentaire les graphiques Graphs. 24,25,26,27 suivants :

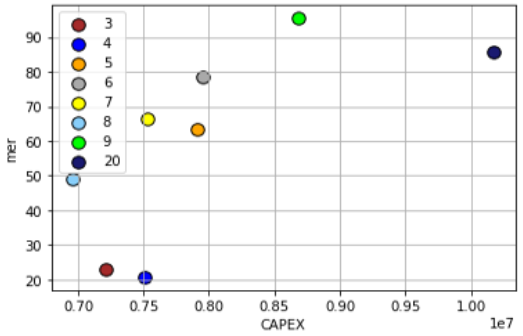


FIGURE 24 – CAPEX en fonction du pourcentage de MER

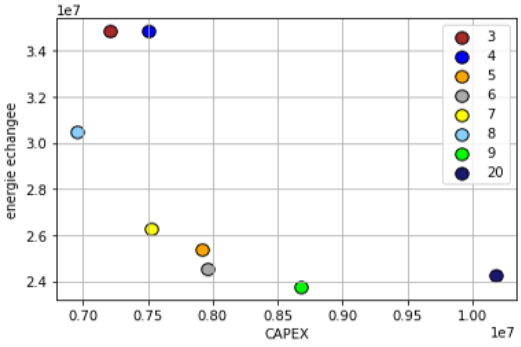


FIGURE 25 – CAPEX en fonction de l'énergie échangée

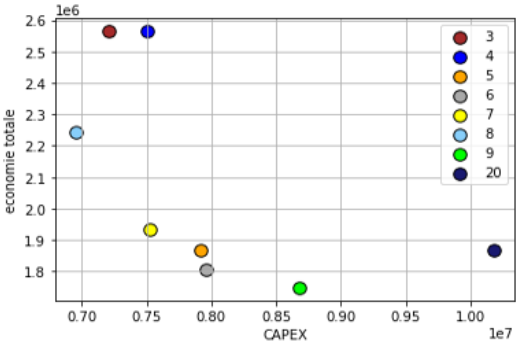


FIGURE 26 – CAPEX en fonction de l'économie totale

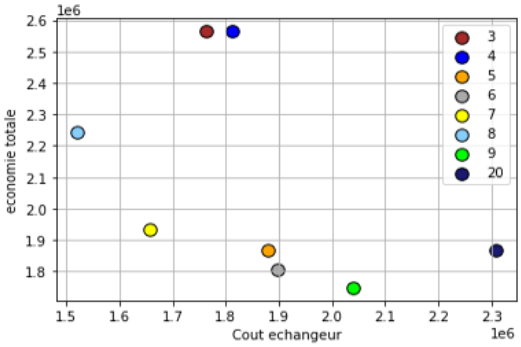


FIGURE 27 – Coût des échangeurs en fonction de l'économie totale

Nous constatons sur le graphique Graph. 24 que le front de Pareto se compose de trois points, le point numéro 3,4 et 8 correspondants respectivement à 3,4 et 8 clusters. Le front de Pareto est une très bonne méthode pour sélectionner des solutions optimales lorsqu'il peut y avoir des conflits, ce qui signifie qu'une solution optimale unique n'existe pas. Dans notre cas, Le point numéro 4 possède le meilleur pourcentage de MER, mais possède une valeur de CAPEX plus élevée que le point numéro 3 et 8. Le point 3 quant à lui possède un meilleur pourcentage de MER que le point 8 mais une valeur de CAPEX plus élevée que celui-ci

et possède une meilleure valeur de CAPEX que le point numéro 4 mais possède un moins bon pourcentage de MER. Finalement, le point 8 possède la valeur CAPEX la plus faible mais possède un moins bon pourcentage de MER que le point 3 et 4. Les autres points sont tous dominés par ces trois points et ne font donc pas parties des solutions optimales. L'étude des cas sur les autres graphiques sont similaires, nous souhaitons maximiser l'économie totale ou l'énergie échangée, nous obtenons alors plutôt deux points optimaux, les points numérotés 3 et 8. De plus, la solution de la méthode statique est dominée par la solution produite avec "8 clusters", elle ne fait donc pas partie des solutions optimales. Par conséquent, en choisissant par exemple la solution proposée par le clustering avec "8 clusters", nous obtenons une diminution de 31 % (resp. 2.4 %) du CAPEX et une augmentation de 20 % (resp. 3.9 %) de l'énergie échangée par rapport aux solutions proposée par la méthode segmentation adaptative (resp. méthode statique). Ainsi, il est intéressant pour les industries d'opter pour ce type de solution.

2.8 Discussion

solutions fournies :

Toutes les solutions par rapport aux performances économiques et énergétiques fournies dans ce rapport sont des solutions théoriques obtenues par le modèle boite-noire du procédé MidRex. La coopération entre l'industrie (procédé MidRex) et Capgemini ayant été close depuis 2020, nous ne pouvons pas vérifier sur le terrain les solutions apportées.

Difficultés rencontrées :

Les difficultés rencontrées se situaient principalement sur la prise en main des algorithmes déjà implémentés par les précédents stagiaires. En particulier, l'utilisation de la boite-noire implémentée en Python fut difficile en raison du code complexe et des contraintes pour pouvoir l'appliquer. Une prise en main de 2 semaines étaient nécessaires avec l'ajout de certaines corrections. La partie clustering quant à elle s'est très bien déroulée et aucune difficulté particulière est apparue.

Documents fournis :

Deux "notebooks" détaillant l'ensemble du travail effectué et du code implémenté ont été ajoutés sur le Gitlab afin de faciliter la prise en main d'une personne tiers. Elles détaillent principalement l'utilisation des algorithmes de clustering utilisés ainsi que certaines figures obtenues. De plus, les fichiers Excels et Word fournissant les résultats de la boite-noire obtenus dans ce rapport ont été conservés et placés dans un dossier dénommé "Result/" sur le gitlab. Un article externe rédigé en anglais a également été rendu afin de pouvoir reprendre mes travaux et de pouvoir publier nos recherches.

2.9 Conclusion :

Au cours de ce rapport, nous avons vu l'enjeu de la récupération de la chaleur fatale dans les industries, notamment du point de vue économique et écologique. nous avons pu alors en premier lieu voir les différentes méthodes de segmentations et de clustering afin d'appliquer la méthode de pincement fournissant notre réseau d'échangeur optimal et ses performances. Un premier choix de la segmentation a été nécessaire et a été basé sur la détection efficace des points de ruptures de notre série temporelle afin d'améliorer les performances du RDSC, incitant ainsi à sélectionner la fonction de coût L_2. De plus, il a été démontré que l'algorithme TICC était 1,64 fois plus efficace selon le critère $F1$-score en comparaison avec l'algorithme RDSC. Dès lors, la méthode de pincement appliquée à l'aide de l'algorithme TICC sur notre boite-noire a permis de confirmer certaines hypothèses telles que l'augmentation financière et la complexité du réseau lorsque le nombre de clusters augmente, mais a permis également d'obtenir des solutions optimales pour les industries. Ces solutions sont très satisfaisantes puisqu'elles permettent aux clients de pouvoir choisir parmi des solutions optimales, celles qui lui correspondent selon son budget et ses attentes. Une solution fournie dans ce rapport

propose par exemple une diminution de 31 % du CAPEX et une augmentation de 20 % de l'énergie échangée par rapport aux solutions proposée par une méthode segmentation adaptative implémentée par les précédents stagiaires.

Malgré les nombreuses recherches effectuées par moi-même ou par les précédents stagiaires pour améliorer la récupération de la chaleur fatale, plusieurs autres études restent ouvertes comme la mise en pratique des résultats obtenus dans ce rapport ou encore l'application de ces méthodes sur d'autres industries.

3 Projet Irradiance solaire

3.1 Contexte :

Face au dérèglement climatique et la nécessité d'apporter des énergies plus propre pour l'environnement, les énergies renouvelables sont devenues un enjeu majeur et font parties des solutions proposées pour satisfaire l'accord de Paris mis en place le 12 décembre 2015. Cette part des énergies renouvelables doit représenter à hauteur de 23 % en 2020 et à 32 % en 2030 de la consommation totale d'énergie finale en France [6]. Actuellement, la part de la production des énergies solaires reste largement limitée dans le monde, représentant seulement 2,1 % de la production totale d'énergie. Cette faible part est souvent dû à des infrastructures solaires complexes et coûteuses et nécessite une connaissance approfondie de l'exploitation des systèmes solaires et de l'optimisation du stockage pour maintenir le niveau de production et répondre à la demande. EDF affirme que cette transition énergétique ainsi que les objectifs associés à la lutte contre le changement climatique amènent les territoires à diversifier leurs sources d'énergie, à intégrer les enjeux d'une production locale de l'énergie et à favoriser un mix énergétique décarboné [6]. L'énergie solaire est une source d'énergie renouvelable qui dépend des paramètres de données d'irradiation pour être efficace. Par conséquent, avant d'investir dans une nouvelle centrale solaire ou tout autres domaines d'applications, il est nécessaire de collecter le plus de données solaires possible. Avec ces données et l'aide de spécialiste, nous obtiendrons une évaluation précise de la ressource solaire et nous pourrons ainsi maximiser les rendements de manière optimale les nombreuses installations liées à l'énergie solaire.

C'est dans ce contexte précis que le projet irradiance solaire a été mis en place afin d'apporter une interface, d'estimer et de prédire l'irradiance solaire au sol partout en France et ceux au temps t sans la nécessité de capteur au sol afin de pouvoir faciliter le développement des infrastructures solaires. Afin d'atteindre nos objectifs, nous nous sommes tournés sur les approches de prédiction à travers les méthodes de machine-learning qui ont su montrer leur efficacité et répondre aux attentes dans ce type de domaine, et en particulier sur des précédentes recherches basées sur la prédiction de l'irradiation solaire.

3.2 Objectif :

Connaissant l'importance de pouvoir estimer l'irradiance solaire au sol partout en France, nous nous sommes focalisés durant cette étude sur l'inférence et la prédiction de cette dernière. En effet, l'avantage d'une telle méthode est qu'elle nécessite seulement des données satellites obtenues par l'organisme EUMETSAT [8] fournit en temps réel avec 11 canaux essentiels, mais également un capteur au sol sur le site de SIRTA [24] afin de connaître la valeur réelle de l'irradiation solaire au sol et ainsi de pouvoir labelliser nos données.

Le projet aura donc pour objectif d'entraîner différents modèles de "machine learning" grâce à ces données pour pouvoir inférer et prédire l'inférence solaire en France dans un horizon de temps court d'une heure ainsi que de proposer une interface graphique aux clients afin de faciliter son utilisation. Par conséquent, une comparaison en profondeur est nécessaire pour sélectionner les meilleurs modèles selon des critères de performances spécifiques. Pour pouvoir mettre à bien notre objectif, il a été convenu de répartir ce projet en 3 parties, la première étant la récupération des données et l'inférence de l'irradiation solaire au sol effectuée par un stagiaire, Matthias Foyer ; la seconde étant la prédiction de l'irradiation solaire au sol effectuée par moi-même, puis enfin la dernière partie consistait à développer une interface graphique conçue par moi-même, Foyer Matthias et Blondel Thomas, un étudiant en alternance dans le développement web. Chacune de ses parties sont très fortement liées et engendre une dépendance et un important travail d'équipe.

Les étapes pour atteindre l'objectif de mes parties ont été organisées de la manière suivante à travers une roadmap :

- ☐ Étape 1 : Reprendre les travaux effectués par Matthias Foyer et s'intégrer dans l'équipe afin de poursuivre l'ensemble des recherches.
- ☐ Étape 2 : Dresser un état de l'art.
- ☐ Étape 3 : Récupérer et analyser les données obtenues.
- ☐ Étape 4 : Appliquer une méthode "forecasting" afin de prédire l'irradiation solaire.
- ☐ Étape 5 : Mettre en place une méthode de partial fit pour contrer la dérivée conceptuelle.
- ☐ Étape 5 : Mettre en place en équipe une application d'inférence et de prédiction afin de faciliter son utilisation.

3.3 Méthodes et approches :

Le calcule l'irradiance solaire de surface (ISS ou SSI en anglais) a incité de très nombreuses recherches en vue de son importance. Des solutions apportées telles que des modèles de transfert radiatif [17] sont envisageables et permettent de calculer cette dernière par des calculs de transfert radiatif dans l'atmosphère. Cependant, il implique certaines contraintes et complexités comme l'acquisition des propriétés spectrales de l'atmosphère et le développement de solveurs d'équations de transfert radiatif qui amène souvent à une solution coûteuse et même irréaliste lors de l'exécution de ces modèles (Modtran ou LibRadtran). L'approche de ces méthodes de transfert radiatif se base donc afin de palier ces problèmes sur la réduction de dépendance spectrale et sur la simplification de la solution de transfert radiatif tout en conservant la précision autant que possible. Des solutions comme le **REST2** [20] fait usage à la fois de l'albédo [5] et des données atmosphériques sont proposées lorsque le ciel est dégagé ou le **FARMS** [30] lorsque le ciel est nuageux, mais ces méthodes doivent être utilisées tout d'abord conjointement à l'aide de connaissance de couverture nuageuse et de propriétés des aérosols, nécessitant donc des critères précis ce qui en fait une solution lourde, non adaptée au cours du temps et pas généralisable sur l'ensemble du territoire français ou mondial. Une seconde problématique de cette approche est la présence de dérèglements, calibrations ou d'autres facteurs imprévisibles qui introduit de l'incertitude à notre modèle est induit une perte élevée de performance.
Il est donc très intéressant de s'intéresser à d'autres approches telles que les algorithmes de machine learning qui sont plus flexibles et peuvent s'adapter à plus de sources de données permettant de généraliser de manière fiable les mesures de SSI. De plus, les techniques de machine-learning sont des outils très puissants pour modéliser différent type de signal (ou des séries temporelles) montrant souvent leur efficacité et peuvent être conçu grâce à des analyses périodiques ou stochastiques pour prédire efficacement la valeur de SSI. Parmi ces techniques, Yunjun Lu [31], Shab Gbemou [22] and Alin Ionescu [2] ont appliqué en 2019 des modèles "Long short-Term memory" (LSTM) pour obtenir des prédictions du SSI avec un horizon de temps estimé d'une heure à 4 heures grâce à une grande capacité de mémoire du modèle LSTM, faisant ainsi souvent partie des modèles les plus efficaces dans de nombreux articles. Nous avons étudié différents types de d'architecture du modèle LSTM, dont certains basés sur plusieurs études qui ont été publiées comme décrites ci-dessus ou d'autres modèles plus classiques. Dans l'article de Aji Prastetya Wibawa [29] daté de 2022, des réseaux de neurones convolutifs 1D ont été adaptés et implémentés pour faire de la prévision de série temporelle, dont plus particulièrement la prévision de l'irradiation solaire au sol dans l'article rédigé par Huaiguang Jiang [11] fournissant des résultats satisfaisants. Dans l'article de Bixuan Gao [9], une autre méthode combinant des modèles de Gated Recurrent Unit et des images de clear sky SSI a été proposée pour prédire la valeur de SSI. Enfin, des modèles hybrides emboitant des réseaux convolutifs et des modèles de LSTM ou GRU ont été étudiés et comparés par les modèles mentionnés précédemment.

5. L'albédo est le pouvoir réfléchissant d'une surface, c'est-à-dire le rapport du flux d'énergie lumineuse réfléchie au flux d'énergie lumineuse incidente.

3.3.1 Données satellites et capteurs au sol :

La première source majeure de donnée collectée provient de l'organisme **EUMETSAT** qui fournisse un ensemble de données complet couvrant la quasi-totalité de la surface terrestre et donc adapté aux calculs de SSI sur le territoire français. Elle est constituée de 11 canaux d'informations brutes déterminant in situ la réflectance de plusieurs longueurs d'onde dispersées sur la surface terrestre par les satellites (particulièrement Sentinel-3). Ces canaux sont très importants pour pouvoir aider nos modèles à prédire la présence d'obstacle dans la progression des longueurs d'ondes émis telle que les nuages. Cet organisme est régulièrement sollicité par des entreprises pour leur grande disponibilité et leur qualité. De plus, ces données sont accessibles de manière presque instantanée grâce à l'utilisation d'API.

Notre seconde source de donnée permettant de labelliser nos données satellites est l'organisme **SIRTA** qui est une banque nationale de données provenant du « Site Instrumental de Recherche par Télédétection Atmosphérique » construit en 2021 sur le campus de l'Ecole polytechnique. De nombreux types de données captés in situ sont disponibles sur des plages de temps larges s'étendant généralement jusqu'au jour même de la récupération des données (données très récentes) et nous fournissent ainsi la valeur réelle d'irradiation solaire de surface sur ce site. Nous avons également ajouté le Clear_sky_GHI qui permet de déterminer l'irradiation solaire au sol si aucun obstacle n'est présent (nuages, pollution, etc ...) afin d'améliorer les performances de l'inférence du SSI.

L'ensemble des données utilisé couvre une période de 2018 à 2021 et a été ré-échantillonné avec un pas de temps $\Delta t = 15$ min, soit un total de 134 493 points. De plus, ces données sont organisées sous la forme de 13 colonnes dans l'ordre suivant : GHI (SSI), Clear_sky_GHI, channel 1, ..., channel 11. Nous avons ensuite découpé nos données avec une répartition de 70 % pour le jeu d'entraînement, 20 % pour le jeu de validation et 10 % pour le jeu de test. Afin que les variables soit mise à la même échelle, nous avons appliqué une normalisation min-max pour que toutes les données se situent dans l'intervalle $[0, 1]$. Cette normalisation est importante pour éviter de donner de l'importance à certaines variables et est calculée comme suit :

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{11}$$

De plus, on peut noter la présence de quelques valeurs manquantes dans nos données s'élevant à 0.8 % de l'ensemble de nos données. Ce nombre étant relativement négligent, l'utilisation d'un système sophistiqué pour les remplacer n'était pas nécessaire, nous les avons alors simplement supprimés. Enfin, selon la littérature [22], une analyse profonde déterminant l'impacte de la suppression ou du maintien des périodes de nuit sur nos jeux de données a été proposée.

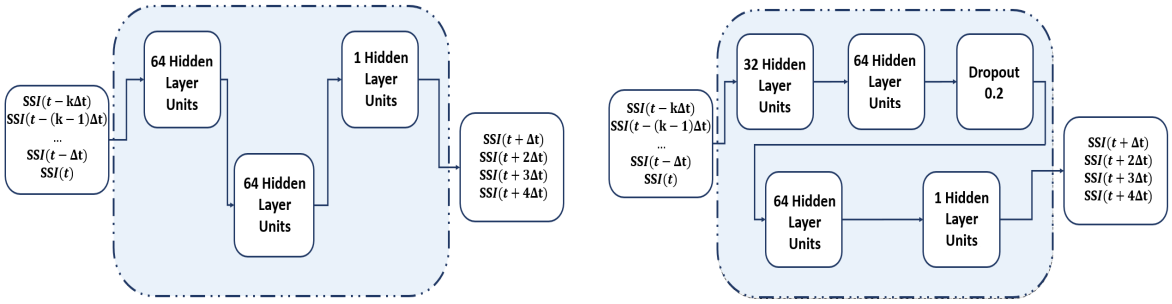
3.3.2 Algorithmes de prédiction du SSI

Notre cas d'étude concerne la prédiction de l'irradiation solaire de surface en France métropolitaine à travers l'utilisation des algorithmes de machine-learning. Basés sur leur réputation et précision, cette section présente les réseaux LSTM (long short-term memory), CNN (convolutional neural network), GRU (gated recurrent unit), RNN (recurrent neural network) et hybrides implémentés avec Tensorflow dont leurs performances seront comparées avec un modèle de référence. Ce modèle de référence est simplement construit de telle sorte à ce que la prédiction du SSI dans une heure soit la valeur de SSI actuelle. Ainsi, cette prédiction n'étant pas si mauvaise, le principe sera alors que les modèles mentionnés ci-dessus aient des meilleures performances que ce modèle de référence. Afin de pouvoir comparer les performances de nos modèles et ceux issus de la littérature, les critères Mean Absolute Error (MAE) et Root Mean Squared Error (RMSE) ont été utilisés. De plus, il est nécessaire d'utiliser le même ensemble de donnée pour chaque modèle entraîné ainsi que le même jeu de validation et de test pour une comparaison juste et équitable.

Dans ce projet, nous avons considéré un « batch size » équivalent à 16 h, soit (64×15 min). L'ensemble de données est mélangé pour s'entraîner sur différentes périodes et les performances des tests et des validations sont mesurées sur l'ensemble de données des tests et des validations. Nous considérons dans la suite de rapport, x et y représentant respectivement l'input et l'output de nos modèles tels que $x_i \in \mathbb{R}^{ws \times n}$ et $y_i \in \mathbb{R}^{4 \times 1}$ avec ws étant une taille de fenêtre de 24 temps, soit un horizon de temps de 6 h, et n étant 13

variables (channel 1, ..., channel 11, Clear sky GHI, GHI) de nos données. À noter que $y_i \in \mathbb{R}^{4\times 1}$ puisque nous voulons prédire seulement la valeur de SSI (1 seule variable) dans un temps futur équivalent à 1 heure, soit 4×15 min. Enfin, un module de "early-stopping" est ajouté dans l'implémentation de nos modèles afin d'arrêter l'entrainement lorsqu'on observe une perte de performance lors de l'évaluation sur le jeu de validation.

Recurrent neural network : Un réseau neuronal récurrent (RNN) est un type de réseau neuronal artificiel utilisé lorsque les utilisateurs souhaitent effectuer des opérations prédictives sur des données séquentielles ou chronologiques, il est très répandu et efficace pour ce type de problème avec l'utilisation de plusieurs couches de réseaux denses afin que ce réseau donne une sortie en fonction de l'entrée précédente et de son contexte. Deux architectures RNN ont été implémentées et illustrées sur la figure Fig. 28

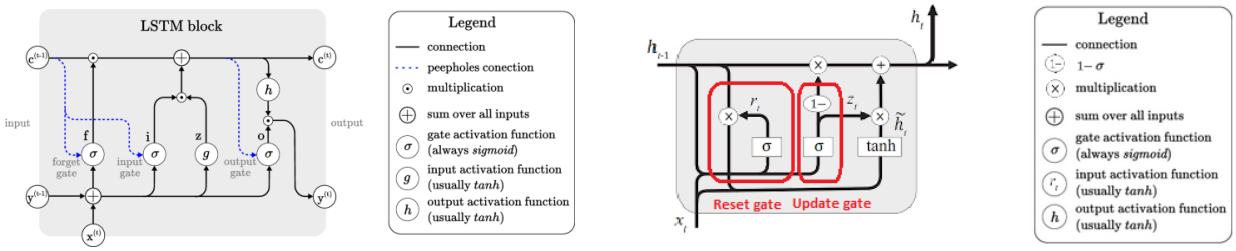


(a) Structure de RNN classique numéro 1 (b) Structure de RNN classique numéro 2

FIGURE 28 – structures des modèles de RNN implémentées pour la prédiction du SSI pour un horizon de temps de $t + \Delta t$ à $t + 4\Delta t$ avec en entrée un horizon de temps de $k\Delta t$ où $\Delta t = 15$ min.

Les deux structures sont relativement simples avec seulement 3 ou 4 couches cachées de neurones avec l'ajout d'un coefficient de droupout pour éviter le sur-apprentissage dans la figure Fig. 28b. Ajouter des couches supplémentaires peuvent générer un réseau trop complexe avec des performances plus faible, par conséquent, une analyse plus profonde n'a pas été envisagée due à une analyse plus centrée sur des modèles réputés plus efficaces dans les réseaux de neurones artificiels tels que le CNN, LSTM ou GRU.

LSTM et GRU : Bien que les réseaux de neurones récurrents soient populaires avec cependant certaines limites telles qu'une courte mémoire par exemple, on s'est intéressé vers de meilleures architectures avec les modèles **LSTM** et **GRU**.Leurs fonctionnements sont basés sur des portes permettant de conserver ou de supprimer des informations qu'il a en mémoire et ainsi de l'utiliser pour pouvoir prédire des temps futurs grâce à une **mémoire à long terme**. Cependant, leurs systèmes de portes diffèrent entre elles dont les principales différences sont présentées sur la figure Fig. 29 ci-dessous.



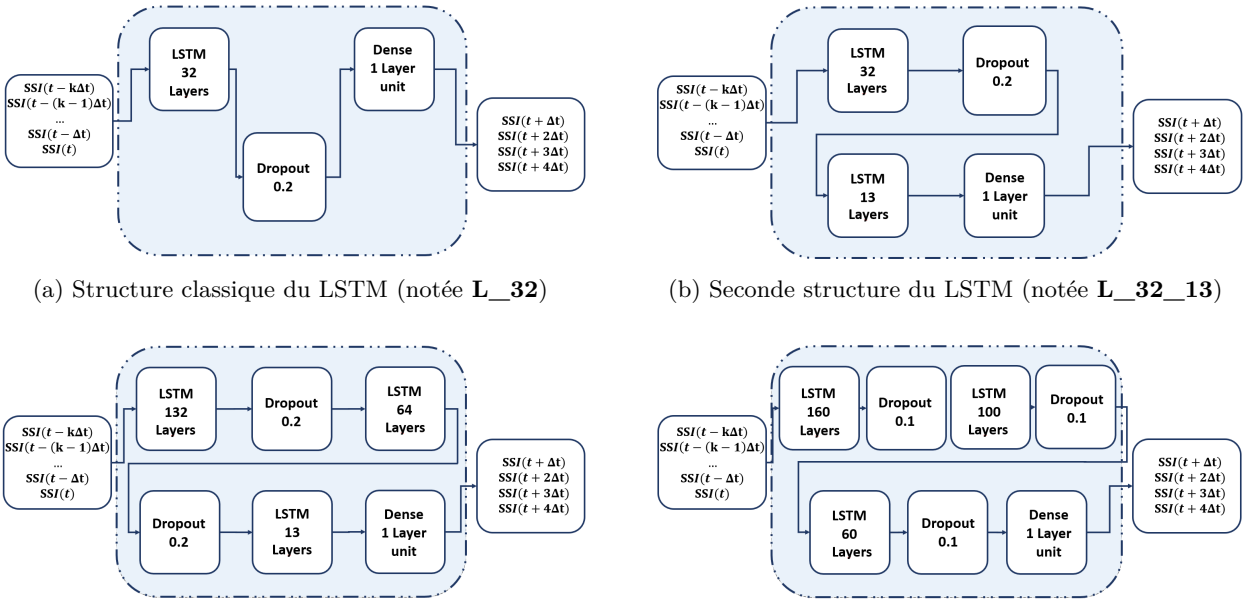
(a) Principe du Long Short-Term memory (b) Principe du Gated Recurrent Unit

FIGURE 29 – Les blocs de fonctionnement des modèles Long Short-Term memory et Gated Recurrent Unit.

Le système GRU comparé au système LSTM n'a pas de cellule d'état (cell state) représentant la connexion entre c^{t-1} et c^t dans la figure Fig. 29a) afin d'assurer la diffusion de l'information initiale. De plus, le système GRU contient seulement deux portes (les portes reset et update) comme illustré dans la figure Fig. 29b) alors que le système LSTM quant à lui en possède trois (les portes "forget", "input" et "ouput"). Un résumé de ces systèmes de porte est expliqué ci-dessous :

- GRU :
  - — la porte reset détermine la quantité d'information à oublier.
  - — la porte update détermine la quantité d'information à maintenir.
- LSTM :
  - — la porte d'oubli (forget gate) détermine les informations à oublier dans la cellule d'état.
  - — Une fois que l'on sait ce que l'on doit oublier, la porte d'entrée (input gate) détermine ce que l'on doit maintenant dans la cellule d'état
  - — La porte de sortie calcule simplement la nouvelle cellule d'état.

De nombreuses architectures LSTM et GRU ont été implémentées mais seulement celles considérées fondamentales sont illustrées dans la figure Fig. 30 pour le système LSTM et Fig. 31 pour le système GRU :



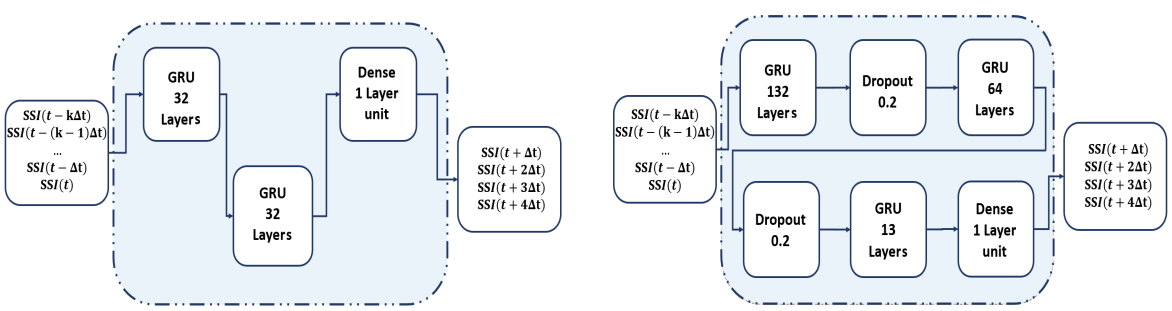
(a) Structure classique du LSTM (notée **L_32**)

(b) Seconde structure du LSTM (notée **L_32_13**)

(c) Troisième structure du LSTM (notée **L_132**)

(d) Quatrième structure du LSTM (notée **L_160**)

FIGURE 30 – Les structures LSTM implémentées pour la prédiction du SSI pour un horizon de temps de $t + \Delta t$ à $t + 4\Delta t$, (a) et (b) sont deux structures classiques, (c) est la structure conseillée par Ionescu Alin [2] et (d) la structure conseillée par Shab Gmebou [22].



(a) Structure de GRU classique (notée **G__32__32**) (b) Seconde structure du GRU (notée **G__132__64**)

FIGURE 31 – structures des modèles GRU implémentées pour la prédiction du SSI pour un horizon de temps de $t + \Delta t$ à $t + 4\Delta t$, (a) une structure classique du modèle GRU et (b) une structure inspirée du modèle (c) du LSTM

Convolutional neural network : Un des algorithmes les plus performants du Deep Learning sont les réseaux de neurones convolutifs ou CNN permettant notamment la reconnaissance d'images en attribuant automatiquement à chaque image fournie en entrée, une étiquette correspondant à sa classe d'appartenance mais qui peuvent s'avérer également très performant sur les séries temporelles en l'ajustant sur un réseau de convolution 1D. Certaines recherches [11, 32] ont utilisé des images provenant de satellites avec les modèles CNN pour détecter la présence de nuage afin de prédire efficacement la valeur de SSI. Cependant, cette méthode peut s'utiliser sur des zones locales telles que des villes mais peut difficilement être utilisée sur un territoire large comme la France. Les avantages du CNN sont dues à sa capacité d'apprendre à travers l'évolution et la forme de la série temporelle ainsi que ces variations. La figure Fig. 32 illustre l'architecture du réseau convolutif implémentée.

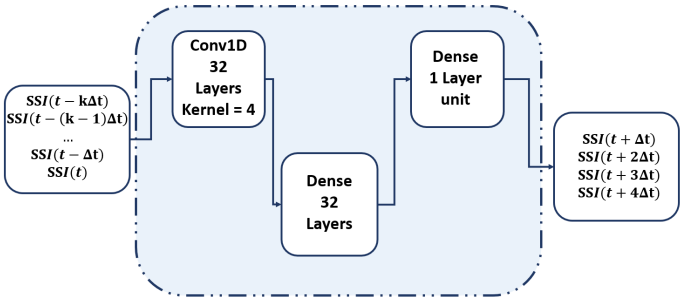


FIGURE 32 – Structure du CNN-1D implémentée (notée **C__32__D__32**)

Une fonction non-linéaire a été ajoutée dans les couches intermédiaires avec l'activation "Relu" afin que le réseau de neurones puisse apprendre plus rapidement et plus facilement par rapport à des couches plus chargées en neurone.

ResNet : Le modèle de référence s'inspire du principe que le SSI ne varie pas considérablement entre deux temps proches consécutifs. Les autres modèles ont été initialisées aléatoirement et ont dû dès lors apprendre que la sortie est un changement relativement faible par rapport au temps précédent. Pour corriger cela, nous allons exiger à notre modèle de prédire uniquement les différences entre deux temps. C'est le principe des réseaux résiduels ou ResNet où chaque couche a un effet additif sur le flot d'information. L'idée se base sur le fait qu'on initialise notre modèle comme le modèle de référence et convergera ainsi plus rapidement, fournissant possiblement des performances légèrement meilleures. La figure Fig. 33 illustre l'architecture du réseau ResNet implémentée.

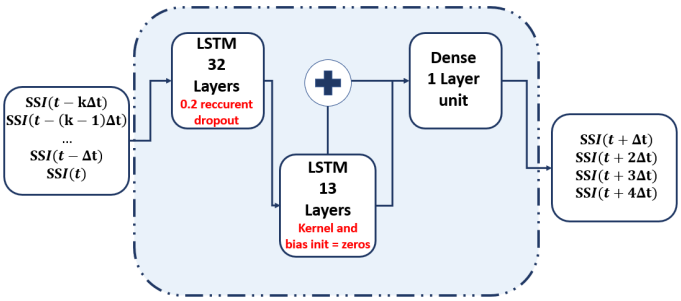
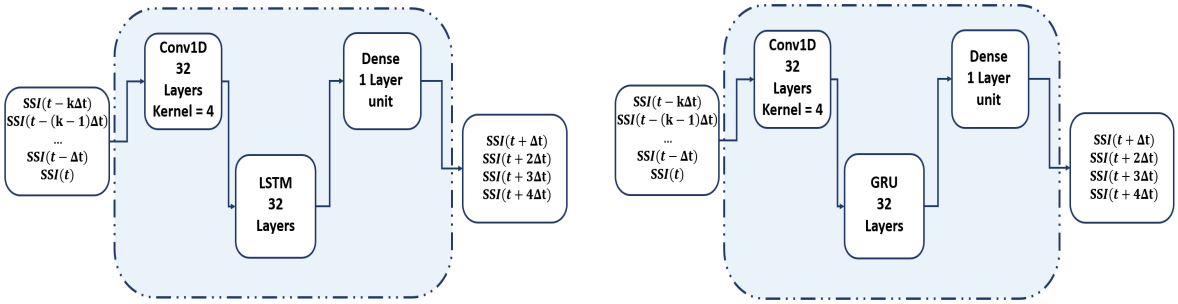


FIGURE 33 – Structure du ResNet implémentée (notée **ResNet**)

Modèles hybrides : Plusieurs articles ont démontré que les modèles hybrides ont de meilleures performances par rapport aux modèles mentionnés ci-dessus. En combinant plusieurs modèles, la précision des prédictions peut être supérieure et gérer certains défauts des modèles précédents tels que des défauts de sur-apprentissage ou de complexité. Dans notre cas, les méthodes hybrides explorées combinent les modèles convolutifs avec les modèles LSTM ou GRU comme conseillé dans les articles [13, 18].



(a) Structure hybride CNN-LSTM notée **C__32__L__32** (b) Structure hybride CNN-GRU notée **C__32__G__32**

FIGURE 34 – structures des modèles hybrides implémentées pour la prédiction du SSI pour un horizon de temps de $t + \Delta t$ à $t + 4\Delta t$, (a) une structure hybride CNN-LSTM et (b) une structure hybride CNN-GRU .

3.3.3 Interface graphique :

Afin de faciliter l'utilisation de l'inférence et de prévision du SSI à nos potentiels clients, une interface graphique a été mis en place. Cette interface a nécessité la collaboration de deux stagiaires, moi-même et Matthias Foyer, ainsi qu'un étudiant en alternance en développement web, Thomas Blondel. Chacun avait un rôle essentiel et bien défini pour mettre en place ce système complexe d'interface. l'ensemble de ces étapes est illustré sur la figure Fig. 35 suivante puis expliqué.

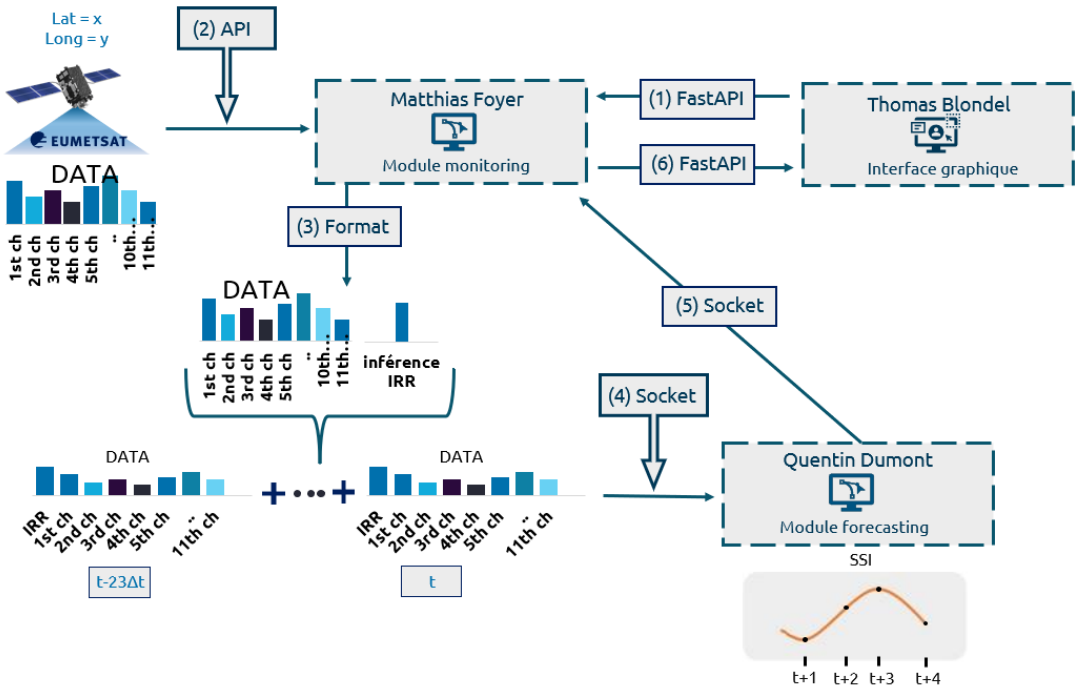


FIGURE 35 – L'ensemble du système complexe pour mettre en place l'interface graphique

La première étape débute par l'interface graphique à travers une page internet qui a été conçue par Thomas Blondel. Sur cette interface, un utilisateur choisi un lieu précis en France métropolitaine (espace maritime compris) en cliquant simplement sur une carte interactive. Cette manipulation permet alors d'obtenir deux données : la latitude et la longitude. Puis, un serveur Web ASGI pour Python a été hébergé à l'aide d'Uvicorn afin que l'interface puisse transmettre et recevoir des données. Ainsi, la longitude et la latitude peuvent alors être transmises sous format json vers le module de monitoring conçu par Matthias Foyer à travers l'utilisation d'un FastAPI étant un framework Web moderne et performante pour la création d'une API sur Python. Ce module de monitoring écoute perpétuellement cette interface, ainsi, dès qu'une nouvelle longitude ou latitude est transmise par notre interface, le module de monitoring le reçoit immédiatement et peut traiter l'information. Dès lors que ce module de monitoring reçoit ces deux données, la seconde étape peut avoir lieu. Cette seconde étape consiste à récupérer à la localisation exacte reçue de l'utilisateur, les données satellites via l'API de l'organisme. Cette étape peut durer quelques secondes-minutes pour extraire toutes les informations. Ces données satellites sont très importantes pour pouvoir inférer la valeur de SSI quelque soit l'endroit en France. Une fois ces données récupérées, un modèle d'inférence dans le module de monitoring permet l'obtention immédiate de l'inférence du SSI. L'obtention de cette dernière permet de passer à l'étape 3 qui consiste à mettre l'ensemble des données nécessaires pour le module de forecasting dans un format adéquat et spécifique. Ce format nécessite un historique de 24×15 min des 11 canaux du satellite Eumetsat, de l'inférence du SSI et du clear sky SSI sous la forme d'une dataframe. Cet historique est possible grâce à un module non présenté sur la figure permettant de mettre à jour un fichier continuellement et indépendamment des autres modules contenant l'historique 24×15 min donnée. Une fois les données mises au bon format, la dataframe contenant toutes les informations nécessaires peut être envoyé grâce à l'utilisation de socket et attendra la réponse du module de forecasting (Étape 4). En effet, le module de monitoring et de forecasting communiqueront uniquement grâce à des sockets dont chacun aura un port d'écoute spécifique (le port numéro 8000 pour le module de monitoring et le port numéro 8002 pour le module de forecasting). Une fois le module de forecasting en possession de la dataframe, un modèle de forecasting l'utilise comme entrée afin de récupérer en sortie les prédictions $t + 15$ min,$t + 30$ min,$t + 45$ min,$t + 1$ h. Ces prédictions sont renvoyées sous le format d'une dataframe vers le module de monitoring qui attend la réponse par socket (Étape 5). Dernière étape, le module de monitoring concatène l'inférence et les prévisions du SSI et les convertis en format json afin d'être envoyé vers l'interface graphique grâce au FastAPI. Ce fichier json est

traité par l'interface pour afficher les différents graphiques.

3.3.4 Dérive conceptuelle :

En Machine Learning, on appelle « **dérive conceptuelle** » [19], les propriétés statistiques de la variable cible que le modèle tente de prédire et qui change avec le temps de manière imprévue. Cela est un problème car les prévisions deviennent moins précises au fil du temps. La dérive conceptuelle est une contrainte courante en fouille de flux de données et dans notre cas, cette dérive peut intervenir lors de dérèglements, des recalibrations de capteurs, lors d'un changement de qualité de l'air ou d'autres facteurs imprévisibles.

Afin de résoudre le problème de dérive conceptuelle, un système de « **partial fit** » est mise en place dès lors que notre modèle perd en performance, ce qui permet de se mettre à jour avec des données plus récentes afin que notre modèle gagne de nouveau en performance mais localement (c'est-à-dire pour des temps futurs court) malgré un risque de perte de performance globale. Plusieurs librairies mettent en place de manière directe ou indirecte ce système de partial fit comme scikit-learn dont certains algorithmes de machine learning possèdent leurs propres fonctions de partial-fit ou Tensorflow dont le système de sauvegarde du modèle (fonction save()) et de rechargement du modèle (fonction load()) permettent l'utilisation d'une mise à jour de l'entraînement du modèle. Cependant, le module du "forecasting" utilise les données inférer par le modèle monitoring afin de prédire l'irradiance solaire comme expliqué précédemment dans la section Section 3.3.3. Or ce modèle de monitoring prend déjà en compte ce système de "partial fit" et corrige donc les potentielles dérives conceptuelles, il n'a donc pas été nécessaire d'ajouter cette étape également sur nos modèles de forecasting.

3.4 Résultats

Dans cette section, nous nous intéressons aux performances des prédictions selon l'impact de certaines variables (nommées feature importance), de la saisonnalité et de la sélection des modèles avec différentes configurations. En particulier, nous voulons connaître, parmi l'ensemble des modèles implémentés, celle qui semble la plus efficace pour la prédiction du SSI selon des critères de performance. Dans ce rapport, deux versions ont été étudiées, dont la première consiste à conserver l'ensemble de notre jeu de donnée lors de l'entrainement des modèles (V1) mais dont la seconde supprimera la période des nuits sur ce même jeu de donnée (V2).

3.4.1 Saisonnalité :

L'étude de la saisonnalité est une étape importante lors de la sélection des modèles, puisque l'absence de saisonnalité dans les données risque de compromettre l'efficacité de certains modèles comme les modèles LSTM ou GRU. En effet, la saisonnalité permet aux réseaux de neurone de s'adapter aux variations de l'irradiation solaire de surface dans le temps et d'apprendre plus efficacement lors de l'entraînement. Par conséquent, une analyse de la saisonnalité est présentée dans la figure Fig. 36 afin de déterminer le niveau de complexité des prédictions du SSI. Comme attendu, une forte saisonnalité quotidienne et annuelle est observée grâce à deux pics de fréquence d'environ 2.1 MHz and 1.6 MHz représentant respectivement la saisonnalité annuel (resp. quotidienne). Ainsi, les modèles LSTM et GRU peuvent être très efficace pour prédire ce type de donnée.

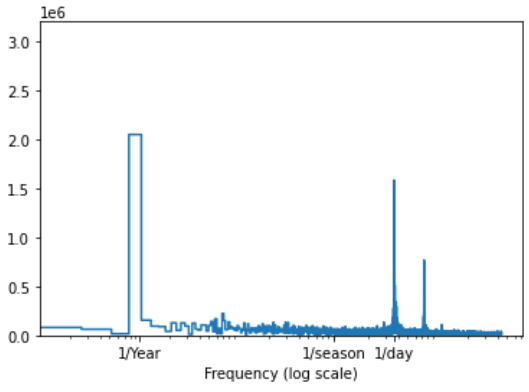


FIGURE 36 – Saisonnalité obtenue à partir des données satellites et du site Sirta couvrant la période 2018-2021

3.5 Feature importance :

Supprimer ou ajouter des informations dans les données peuvent diminuer la performance des modèles, ou à l'inverse, améliorer cette dernière. Déterminer les feature importance pour sélectionner celles ayant le plus d'importance est basée sur une méthode plutôt simple consistant à dériver la sortie du modèle selon chaque variable. Le principe de cette méthode est brièvement expliqué ci-dessous :
Considérons un modèle M tel que :

$$M : \mathbb{R}^T \times \mathbb{R}^N \longmapsto \mathbb{R}^S \times \mathbb{R}^1$$

où T est l'horizon de temps (24 dans notre cas), N le nombre de variables (13 ici) and S l'intervalle de temps prédit (4 ici).
La feature importance est alors calculée comme suit :

$$G_i = \sum_{t=0}^{T} |\partial_{x_{t,i}} M| \tag{12}$$

où $0 \leq i \leq N$.

La figure Fig. 37 illustre les résultats obtenus du modèle LSTM mentionné ici [30c].

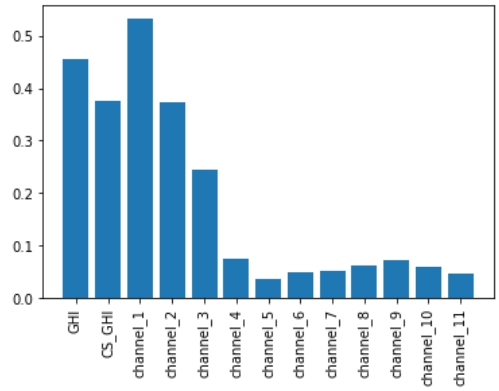


FIGURE 37 – Feature importance du modèle [30c] implémenté

On observe que le GHI représentant l'irradiation solaire de surface, CS_GHI et les trois premiers canaux sont d'après la figure les variables les plus importantes pour la prédiction du GHI (SSI). ces résultats s'expliquent par le fait que le GHI et CS_GHI donne des informations directes sur l'historique récente du GHI, ce qui est très utile pour notre modèle. Les trois premiers canaux, quant à eux, représentent les canaux ayant les longueurs d'ondes les plus élevées émis depuis le satellite, donc ayant les photons les plus faibles en énergie, ce qui le rend plus sensible à la présence d'obstacle comme les nuages. Or, la présence d'obstacle affecte grandement la valeur du GHI puisque cette valeur diminue avec la présence de nuage. On observe qu'il y a des différences significatives sur le reste des canaux étant moins importants pour le modèle mais ont été néanmoins conservés pour d'une part des raisons pratiques dans le développement de notre interface, mais d'autre part, ces canaux ajoutent toujours de l'information utile malgré sa faible importance pour notre modèle.

3.6 Performance de précision des modèles :

Deux versions ont été explorées pour la prédiction du SSI. La deuxième version est basée sur la suppression des périodes de nuit dans nos données pour éviter le sur-apprentissage comme expliqué dans des recherches récentes. Cependant, cette suppression peut générer une interface de monitoring trop complexe et peu flexible lorsque nous voulons une prédiction en tout temps (nuits inclues). En effet, nous avons construit une interface pour qu'elle puisse prédire quelque soit l'heure la valeur du GHI sur l'ensemble du territoire français ainsi

que sa prédiction dans une heure. Dès lors, il nous a semblé essentiel de pouvoir conserver les périodes de nuit dans notre jeu de donnée. Nous avons donc ajouté une étude supplémentaire sur les performances de précision de nos modèles sans et avec les nuits afin de les comparer et de prendre une décision adéquate. Pour des raisons de simplicité, nous introduisons quelques notations dans les résultats :

Notation	
Notation	**Définition**
L	Long Short-Term Memory
G	Gated Recurrent Unit
C	Convolutional Neural Network
D	Dense Network

3.6.1 Version 1 : Conservation des périodes de nuit

Le tableau Table 1 montre les résultats obtenus des performances selon les métriques MAE et RMSE sur nos jeux de test sur l'ensemble des données. Nous rappelons également que le modèle de référence noté baseline retourne simplement la valeur de SSI au temps t pour prédire le temps $t + 1h$.

Accuracy Performance Metrics		
Model	**MAE**	**RMSE**
Baseline	0.0381	0.0805
C_32_G_32	0.0319	0.0552
C_32_D_32	0.0302	0.0571
ResNet	0.0268	0.0548
C_32_C_64_D_32	0.0254	0.0559
L_32_13	0.0248	0.0541
L_160	0.0234	0.0545
C_16_C_32_D_32	0.0227	0.0539
G_132	0.0224	0.0545
C_32_L_32	0.0221	**0.0532**
L_32	0.0218	0.0534
L_132	**0.0214**	0.0539

TABLE 1 – valeurs de MAE et RMSE pour chaque modèle inclus dans notre étude comparative. Pour comprendre les abréviations des modèles, elles sont indiquées sur les structures des modèles dans la section Section 3.3.2

On observe sur la table Table 1 que deux modèles sont très performants selon les métriques MAE et RMSE. En effet, le modèle étant le plus efficace dans les prédictions selon la métrique MAE est le modèle **L_132** [30c] représentant un modèle LSTM avec une amélioration de précision de 44 % par rapport à notre baseline. Selon la métrique RMSE, le modèle le plus efficace dans les prédictions est le modèle **C_32_L_32** [34a] représentant cette fois-ci un modèle hybride avec une amélioration de précision de 33 % par rapport à notre baseline. Ces performances sont très satisfaisantes sachant que le modèle de référence qui consiste simplement à retourner la valeur de SSI au temps t pour prédire le temps $t + 1h$ n'est relativement pas une si mauvaise prédiction. Nous observons également que tous les modèles présentés dans ce tableau ont tous de meilleure performance que la baseline, prouvant ainsi l'efficacité des méthodes de machine-learning. La figure Fig 38 illustre la prédiction de SSI avec un horizon d'une heure pour le modèle **L_132** (courbe orange) comparé aux valeurs réelles du SSI (courbe bleue) sur une durée de 50 h.

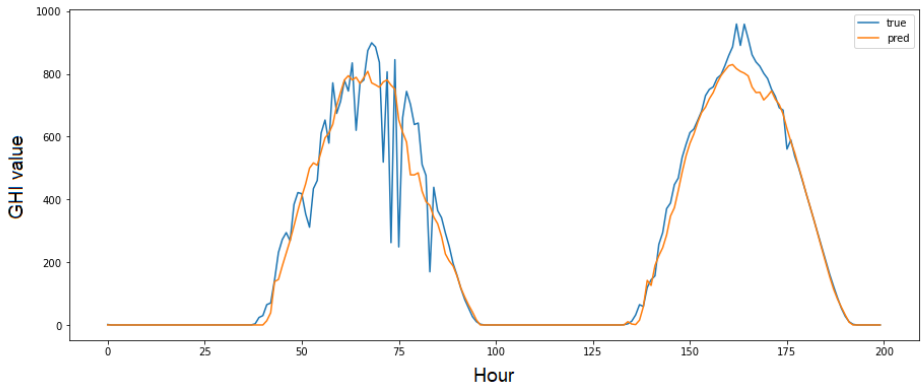


FIGURE 38 – Prédiction de SSI avec un horizon d'une heure sur le site de Sirta du modèle **L_132** (courbe orange) comparé aux valeurs réelles du SSI (courbe bleue) sur une durée de 50 h.

3.6.2 Version 2 : Suppression des périodes de nuit

Dans cette section, les périodes de nuits sont supprimées dans notre jeu de donnée, et nous réitérons les analyses de performance des modèles selon les mêmes métriques précédentes afin de comparer les deux versions. Comme dans la section précédente, nous affichons sur le tableau Table 2 les résultats obtenus des performances selon les métriques MAE et RMSE sur nos jeux de test sur le nouveau jeu de donnée.

Accuracy Performance Metrics		
Model	**MAE**	**RMSE**
Baseline	0.0898	0.131
ResNet	0.0619	0.0936
C_32_C_64_D_32	0.0598	0.0937
L_32_13	0.0587	0.0927
L_132	0.0582	0.0919
C_16_C_32_D_32	0.0572	0.0918
C_32_D_32	0.0570	0.0906
L_32	0.0565	0.0907
C_32_G_32	0.0561	0.0904
G_132	0.0577	0.0923
L_160	0.0554	0.0907
C_32_L_32	**0.0499**	**0.0897**

TABLE 2 – valeurs de MAE et RMSE pour chaque modèle inclus dans notre étude comparative sans les périodes de nuit. Pour comprendre les abréviations des modèles, elles sont indiquées sur les structures des modèles dans la section Section 3.3.2.

Nous observons que seul le modèle **C_32_L_32** est le plus performant selon les deux métriques avec une amélioration de 44 % (resp. 33 %) pour la métrique MAE (resp. la métrique RMSE) par rapport à la baseline. Ces améliorations par rapport à la baseline sont plutôt similaires que celles obtenues sur l'analyse comparative de la section avec les nuits. Cependant, pour faire une comparaison juste et équitable (puisqu'il est évident que la version 1 avec le jeu de donnée complet produit des meilleurs résultats dû à la facilité de prédire la valeur de SSI quand il fait nuit), nous avons décidé de retirer lors du calcul des métriques dans la version 1, la période des nuits et nous avons obtenu une valeur de RMSE de 0.076 pour le modèle L_132. Soit une réduction de 14 % par rapport aux meilleurs modèles de la version 2 (C_32_L_32). Cela s'explique par le fait que malgré que nous ayons supprimé la période des nuits seulement sur le calcul des métriques pour que la comparaison soit équitable, la version 1 possède encore l'avantage de posséder, en entrée des modèles,

tout l'historique précédent des 24×15 min) avec les nuits incluses, aidant ainsi le modèle dans ces prédictions durant la journée (et principalement le matin). La version 2 n'a pas cet avantage puisque l'entrée des modèles possèdent l'historique des 24×15 min mais sans les nuits (les dernières heures de la période ensoleillée du jour précédent sont rattachées aux nouvelles heures de la période ensoleillée du jour actuel). Par conséquent, nous avons conclu que la meilleure alternative pour la prédiction du SSI dans l'interface graphique était la version 1 dont le choix du modèle s'est porté sur le modèle L_132 ayant obtenu les meilleures performances selon la métrique MAE.

3.7 Augmentation de l'horizon de temps pour la prédiction :

Afin de contrôler la précision des prédictions du SSI qui ont généralement des variations relativement élevées, un horizon de temps court d'une heure a été nécessaire. Un horizon de temps plus long engendra en effet une perte de précision comme observée sur la figure Fig 39 suivante :

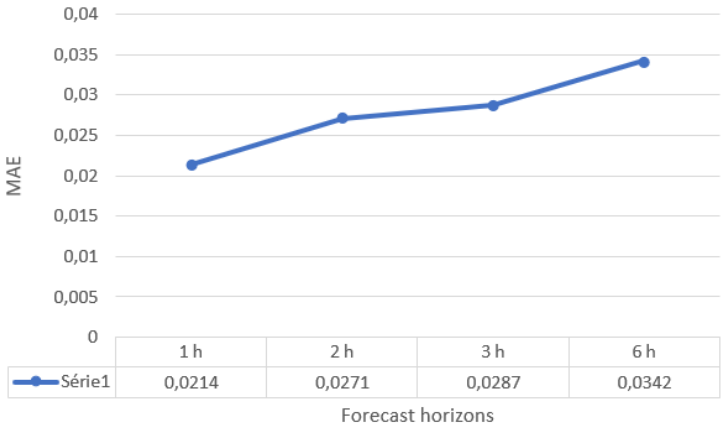


FIGURE 39 – Augmentation de l'horizon de temps pour la prédiction sur le modèle **L_132** avec le jeu de donnée complet

Nous voyons une perte de performance d'environ 37 % selon la métrique MAE pour le modèle **L_132** entre un horizon de temps d'une heure et six heures, ce qui est une perte importante. De ce fait, il nous a semblé approprié de conserver l'horizon de temps d'une heure afin de maintenir une précision élevée de nos prédictions.

3.8 Discussion

Difficultés rencontrées :

Certaines difficultés ont été rencontrées au cours de ce projet. La première étant la récupération des données qui a été une tâche colossale due aux systèmes complexes de l'API des deux organismes. Cette difficulté a également engendré la récupération des mauvaises données de SSI qu'on a pu constater seulement quelques semaines plus tard, impliquant une nécessité de reproduire de nouveaux résultats. La seconde difficulté était le travail collectif pour produire l'interface graphique. En effet, de nombreuses contraintes ont été observées au cours de la mise en place du système, engendrant régulièrement des modifications de chaque module de l'équipe. Il était crucial que chacun modifie sa partie afin de satisfaire toutes les contraintes de tous les modules. Cependant, au vu du nombre de tâches important de chacun, plusieurs jours étaient parfois nécessaires afin que tous les modules soient prêts pour être testés. Néanmoins, une très belle coordination et une très bonne ambiance dans l'équipe ont permis de surmonter ces difficultés.

Documents fournis :

Plusieurs "notebooks" détaillant l'ensemble du travail effectué et du code implémenté ont été ajoutés sur le Gitlab afin de faciliter la prise en main des prochaines équipes de ce projet. Elles détaillent principalement les algorithmes et l'ensemble des résultats obtenus. Un article externe rédigé en anglais a également été rendu afin de pouvoir reprendre mes travaux et de pouvoir publier nos recherches.

3.9 Conclusion

Au cours de ce rapport, nous avons vu l'enjeu de la prédiction de l'irradiation solaire dans les infrastructures solaires avec un point de vue économique et écologique. Nous avons pu suivre les nombreuses étapes d'implémentation des algorithmes ainsi que la mise en place d'une interface graphique pour faciliter l'utilisation de ces prédictions. Un premier choix du modèle à utiliser était nécessaire pour garantir la précision des prédictions de notre interface et a été basé selon deux critères, le MAE et RMSE étant deux métriques largement utilisées dans la littérature. De plus, il a été démontré que deux modèles étaient très performants pour la prédiction, le modèle LSTM **L__132** [30c] avec une amélioration de précision de 44 % par rapport à notre baseline selon la métrique MAE et le modèle hybride **C__32__L__32** [34a] avec une amélioration de précision de 33 % par rapport à notre baseline selon la métrique RMSE. Ces analyses de performance ont permis la mise en place de l'interface graphique avec le choix du modèle LSTM. Cependant, malgré les nombreux travaux effectués au cours de ce projet, plusieurs autres études restent ouvertes comme l'ajout de différents modèles dans l'analyse de comparaison afin de trouver d'autres solutions de prédiction, mais également l'ajout de nouvelles données afin d'améliorer les performances ou encore un développement plus sophistiqué de notre interface graphique.

4 Conclusion

Pour conclure ce rapport de stage, j'ai eu l'opportunité d'être impliqué dans deux projets différents au sein de Capgemini Engineering, me permettant ainsi d'enrichir considérablement mes connaissances aussi bien dans mes deux domaines que sont l'optimisation et la science des données que dans d'autres domaines associés à mes projets tels que la thermodynamique, le système réseau et de nombreux autres domaines. Un atout précieux de mon stage a été la possibilité de travailler en équipe avec un stagiaire et un alternant lors du projet de l'irradiation solaire puisque d'une part, j'ai pu découvrir les bienfaits que procurent la réalisation d'un projet en équipe avec les hautes exigences associées, puis d'autre part, j'ai pu connaître avec mes collègues la satisfaction d'avoir conçu une interface graphique afin d'avoir un rendu visuel de nos longues recherches internes. Les bénéfices de ce stage ne s'arrêtent pas seulement aux points cités précédemment puisque j'ai pu notamment accroître mes expériences professionnelles au sein d'une entreprise leader en conseil d'ingénierie mais également de développer mes expériences sociales en ayant eu l'occasion de discuter longuement avec un grand nombre de personnes au sein du groupe. Enfin, ce stage s'est déroulé parfaitement tout au long de ces 6 mois en compagnie de mon tuteur, stagiaires et collègues, fournissant des résultats très satisfaisants dans le cadre de mes missions. Fort de cette expérience, j'ai l'occasion désormais d'évoluer au sein du groupe dans le cadre de différentes missions que proposent les métiers de consultant.

5 Annexe :

5.1 Bottom-up Segmentation :

Détaillé dans le Rapport de Brice Lamil.

5.2 SVD et PCA :

Soit $X_i \in [a_i, b_i]$ un segment noté $S(a_i, b_i)$ et F_i la matrice de covariance de X_i.
Décomposition en valeur singulière : $X_i = U_i \Sigma_i V_i^T$ où :

— Σ_i c'est la matrice contenant les valeurs singulières en ordre décroissant

— U_i et V_i son changement de base associés aux valeurs singulières.

Analyse en composantes principales : $F_i = V_i \Gamma_i V_i^T$ où :

— Γ_i contient les valeurs propres de F_i en ordre décroissant.

— V_i les vecteurs propres de F_i.

5.3 Matrice de Toeplitz

Nous ajoutons une contrainte sur les θ_i pour être de la forme bloc Toeplitz. Chaque matrice $nw \times nw$ peut être exprimée de la forme suivante :

$$\theta_i = \begin{bmatrix} A^{(0)} & (A^{(1)})^T & (A^{(2)})^T & ... & ... & (A^{(W-1)})^T \\ A^{(1)} & A^{(0)} & (A^{(1)})^T & ... & ... & ... \\ A^{(2)} & A^{(1)} & ... & ... & ... & ... \\ ... & ... & ... & ... & (A^{(1)})^T & (A^{2)})^T \\ ... & ... & ... & A^{(1)} & A^{(0)} & (A^{(1)})^T \\ A^{(w-1)} & ... & ... & A^{(2)} & A^{(1)} & A^{(0)} \end{bmatrix}$$

où $A^{(k)} \in \mathbf{R}^{nw \times nw}$ pour $k \in \{0, w-1\}$ et A_{ij}^k est composé des valeurs de corrélations entre le capteur i au temps t et le capteur j au temps $t+k$.

5.4 Bayesian information criterion :

L'algorithme TICC possède un total de 4 paramètres afin d'obtenir un modèle : le nombre de clusters, la fenêtre w et les coefficients α et β. Il est alors important de pouvoir sélectionner le meilleur modèle TICC, pour cela intervient le critère d'information bayésien (BIC) qui est un critère de sélection de modèle parmi un ensemble fini de modèles. Il est défini de la manière suivante :

$$BIC = -2\ln(L) + k \cdot \ln(N) \tag{13}$$

où L la vraisemblance du modèle estimée, N le nombre d'observations dans l'échantillon et k le nombre de paramètres libres du modèle. À noter que plus la valeur de BIC est faible et plus le modèle est généralement préféré.

5.5 F1-score

Afin de pouvoir comparer l'algorithme TICC et l'algorithme RDSC, nous avons décidé d'utiliser le F1-score [3].
Définition :

Le F1-score est l'une des mesures courantes pour évaluer l'efficacité d'un classificateur. Il nécessite un classificateur de référence et permet ainsi de comparer ce que notre modèle prédit par rapport à notre référence. Il est calculé de la manière suivante :

$$F1 - score = \frac{\sum_{n=1}^{k} F1 - score(n)}{k} \tag{14}$$

où k est le nombre de clusters et avec :

$$F1 - score(n) = \frac{2 \times Precision(n) \times Recall(n)}{Precision(n) + Recall(n)} \tag{15}$$

et :

$$Precision(n) = \frac{TP(class = n)}{TP(class = n) + FP(class = n)} \tag{16}$$

$$Recall(n) = \frac{TP(class = n)}{TP(class = n) + FN(class = n)} \tag{17}$$

Imaginons que notre jeu de donnée est la suivante :

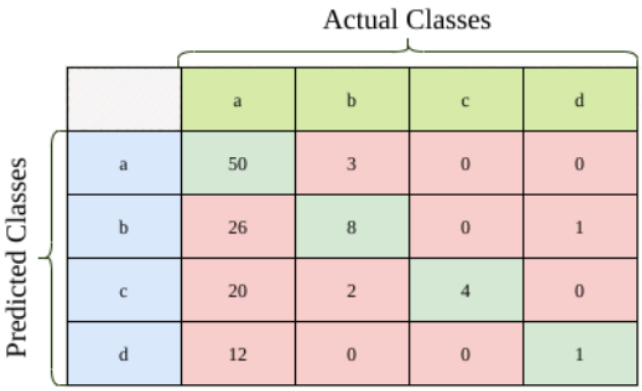


FIGURE 40

Calculons maintenant le F1-score pour la première classe, qui est la classe n . Nous devons d'abord calculer les valeurs de précision et de recall :

$$Precision(n) = \frac{TP(class = n)}{TP(class = n) + FP(class = n)} = \frac{50}{50 + 3} = 0.943 \tag{18}$$

$$Recall(n) = \frac{TP(class = n)}{TP(class = n)} + FN(class = n) = \frac{50}{50 + (26 + 20 + 12)} = \frac{50}{108} = 0.463 \tag{19}$$

$$F1 - score(n) = \frac{2 \times Precision(n) \times Recall(n)}{Precision(n) + Recall(n)} = \frac{2 \times 0.943 \times 0.463}{0.943 + 0.463} = 0.621 \tag{20}$$

Il suffit ensuite de faire varier les valeurs de n possibles pour obtenir le F1-score final.

5.6 Ensemble des flux :

Flux chaud	Gaz de refroidissement	T_e	coolingGasOffTakeA.T
		T_s	air
		Debit	diff_coolingGasln_CZ_NG_FR
	Eau de refroidissement du compresseur top gas	T_e	None
		T_s	None
		Debit	None
	Eau de refroidissement du compresseur cooling gas	T_e	None
		T_s	None
		Debit	None
	Gaz d'échappement du reformer A	T_e	flueGasToStackA_T
		T_s	air
		Debit	coldFuelGas_FR
	Gaz d'échappement du reformer B	T_e	flueGasToStackA_T
		T_s	air
		Debit	coldFuelGas_FR
Flux froids	Apport du gaz naturel au reformer A	T_e	T_NG
		T_s	ProcessGasA_T
		Debit	processNGtotA_FR
	Apport du gaz naturel au reformer B	T_e	T_NG
		T_s	ProcessGasA_T
		Debit	processNGtotA_FR
	Apport du gaz naturel au reducing gas	T_e	EnrichmentNG.T
		T_s	bustleGas
		Debit	enrichmentNG_FR
	Apport d'air au reducing gas	T_e	air
		T_s	bustleGas
		Debit	enrichmentNG_FR
	Gaz combustible du reformer A	T_e	coldFuelGasA_T
		T_s	bustleGas
		Debit	coldFuelGas_FR
	Gaz combustible du reformer B	T_e	coldFuelGasA_T
		T_s	bustleGas
		Debit	coldFuelGas_FR
	Apport de gaz naturel au Fuel_A	T_e	mainBurnerNG.T
		T_s	bustleGas
		Debit	burnerNGA_FR
	Apport de gaz naturel au Fuel_B	T_e	mainBurnerNG.T
		T_s	bustleGas
		Debit	burnerNGA_FR
	Apport auxiliaire de gaz naturel au Fuel_A	T_e	auxNG.T
		T_s	bustleGas
		Debit	auxNGA_FR
	Apport auxiliaire de gaz naturel au Fuel_B	T_e	auxNG.T
		T_s	bustleGas
		Debit	auxNGA_FR
	Apport auxiliaire d'air au Fuel_A	T_e	auxAir.T
		T_s	hotCombustionAirA_T
		Debit	processGas_plus_processNGtot_FR
	Apport auxiliaire d'air au Fuel_B	T_e	auxAir.T
		T_s	hotCombustionAirA_T
		Debit	processGas_plus_processNGtot_FR

5.7 Référence de clustering

Les figures suivantes montrent [fig 41,45] les références de clustering en fonction du nombre de clusters.

3 clusters

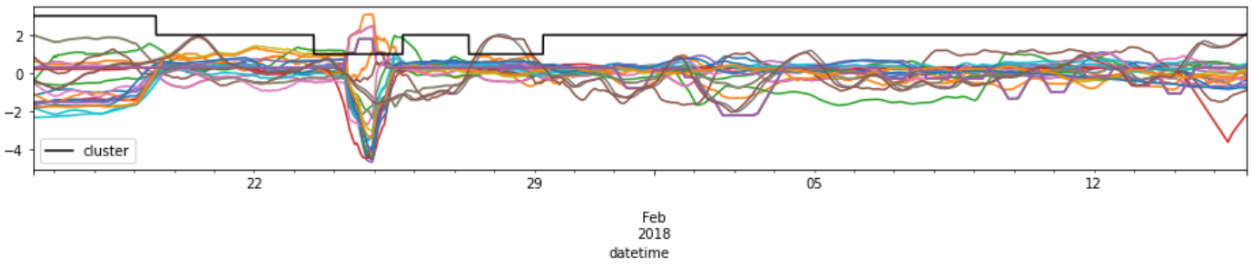


FIGURE 41 – Clustering de référence obtenu par l'annotateur 1

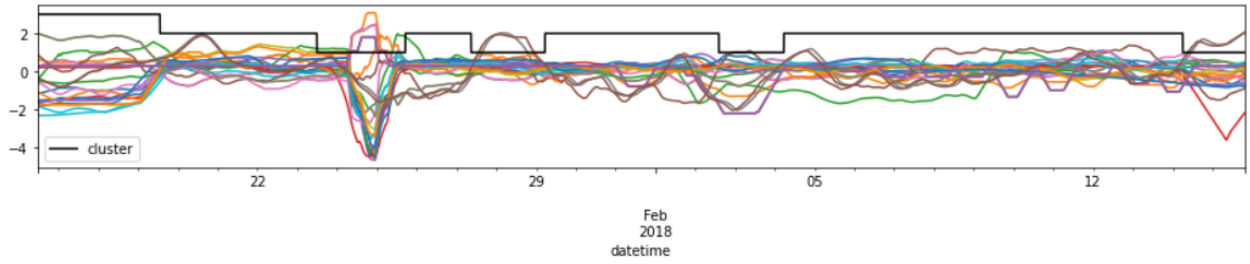


FIGURE 42 – Clustering de référence obtenu par l'annotateur 2

4 clusters

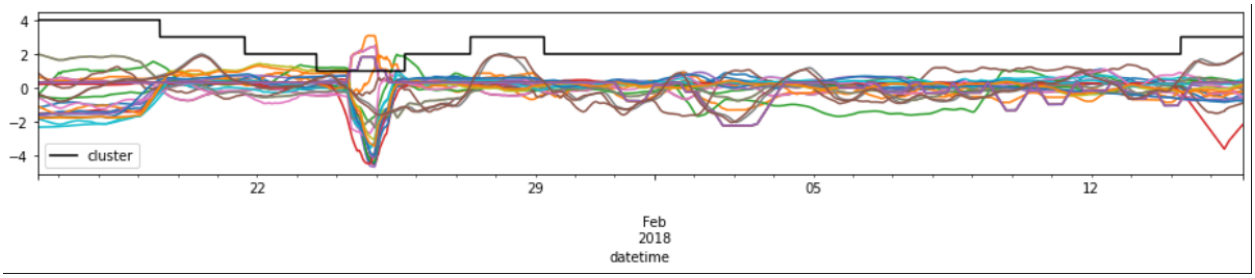


FIGURE 43 – première version du clustering de référence obtenu par l'annotateur 1

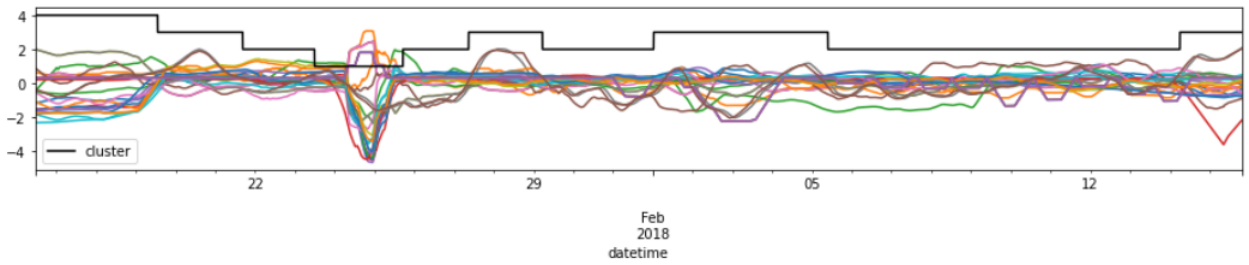


FIGURE 44 – deuxième version du clustering de référence obtenu par l'annotateur 1

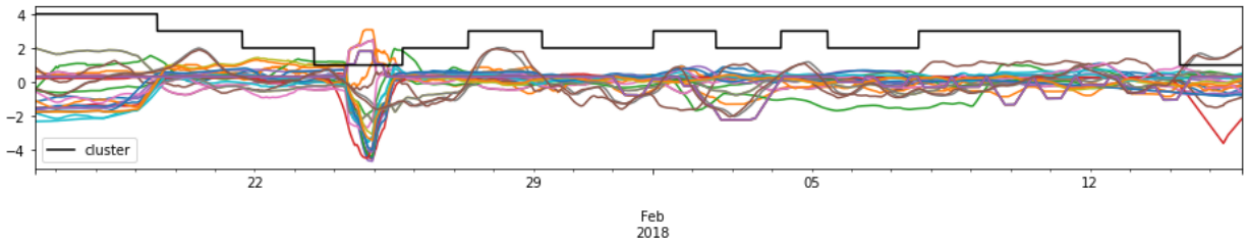


FIGURE 45 – Clustering de référence obtenu par l'annotateur 2

---

5 clusters

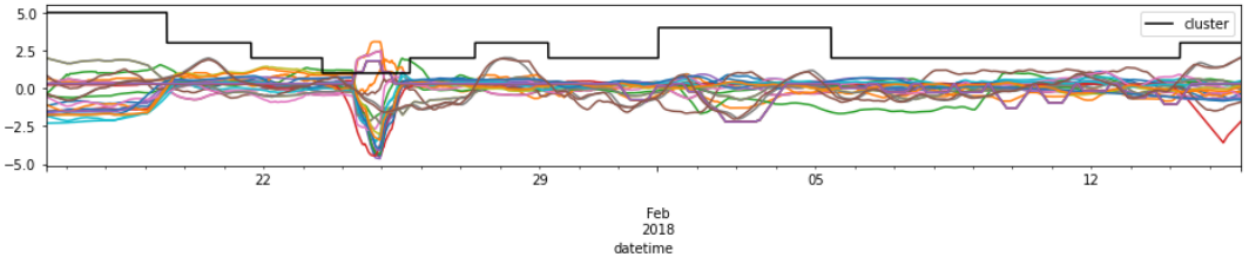


FIGURE 46 – première version du clustering de référence obtenu par l'annotateur 1

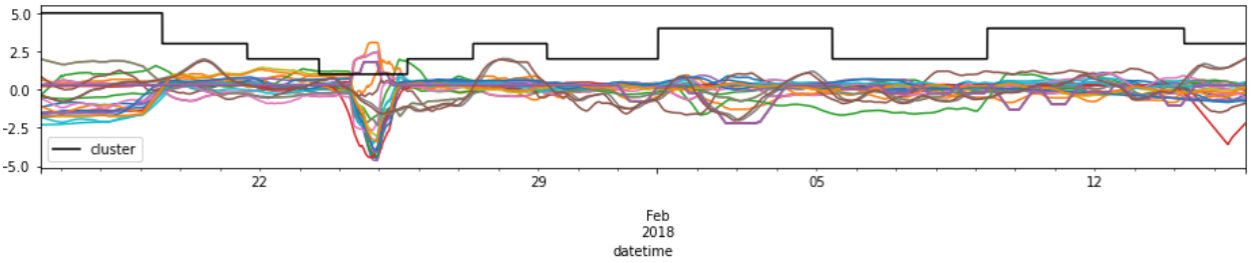


FIGURE 47 – deuxième version du clustering de référence obtenu par l'annotateur 1

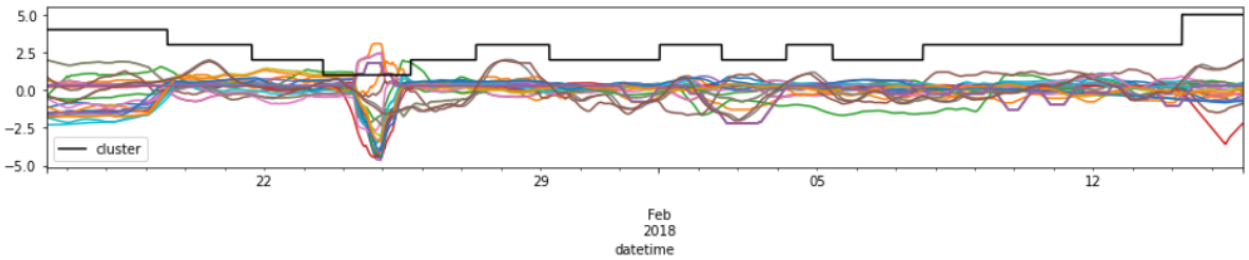


FIGURE 48 – Clustering de référence obtenu par l'annotateur 2

5.8 Résultat de l'algorithme TICC et RDSC

Les figures suivantes montrent les résultats de nos deux algorithmes pour les différentes valeurs de clusters :

---

3 clusters

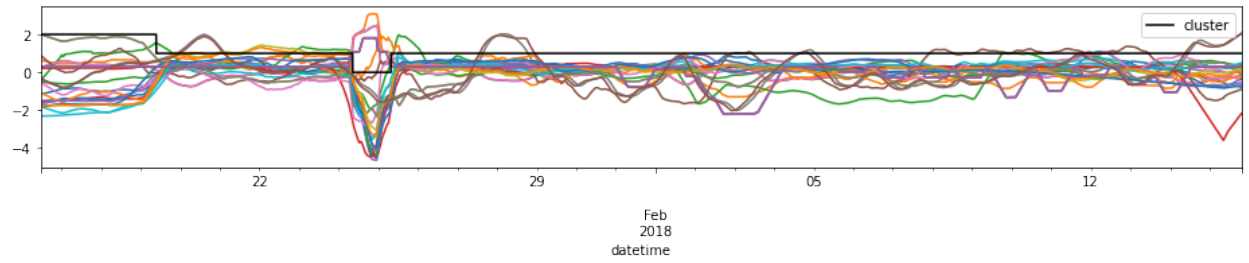


FIGURE 49 – Clustering obtenu par l'algorithme TICC

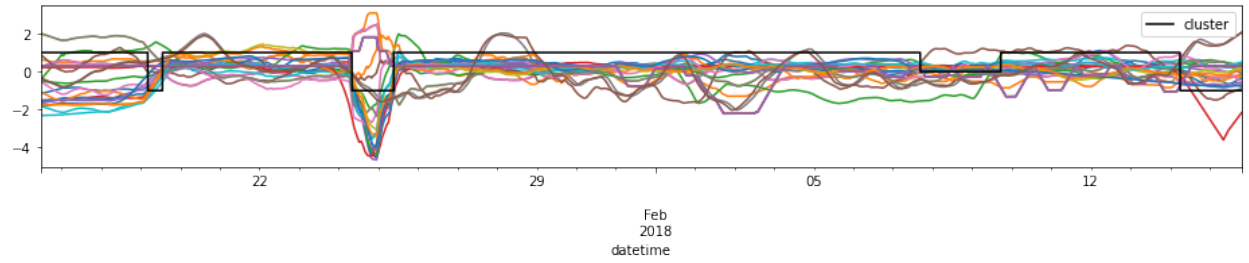


FIGURE 50 – Clustering obtenu par l'algorithme RDSC

---

4 clusters

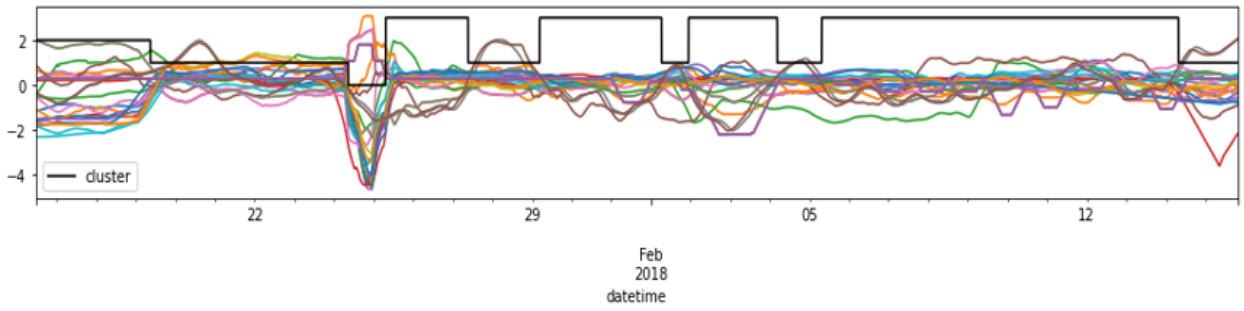


FIGURE 51 – Clustering obtenu par l'agorithme TICC

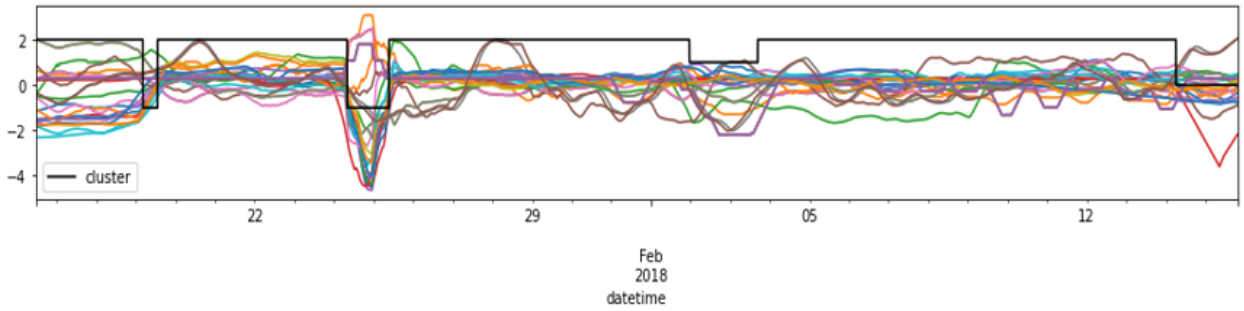


FIGURE 52 – Clustering obtenu par l'agorithme RDSC

---

5 clusters

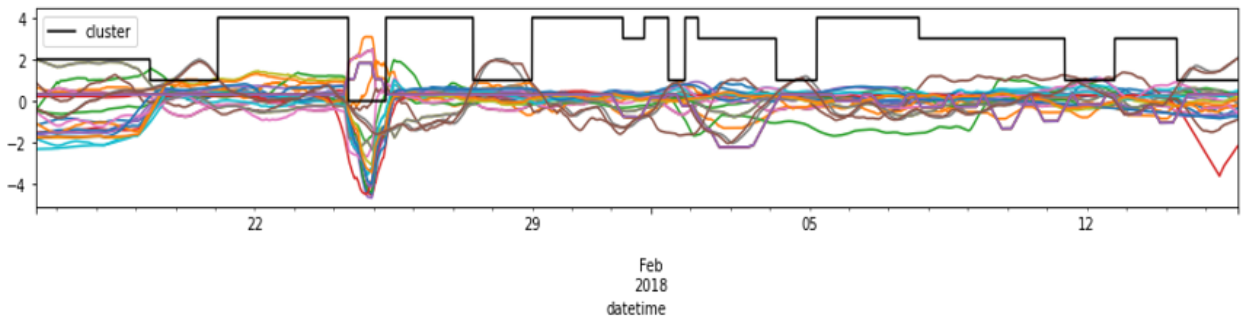


FIGURE 53 – Clustering obtenu par l'agorithme TICC

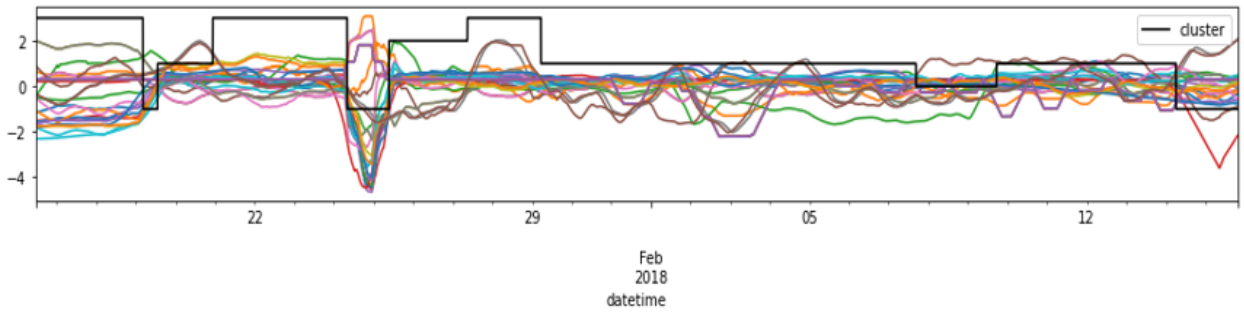


FIGURE 54 – Clustering obtenu par l'agorithme RDSC

Références

[1] Abonyi. "P. Principal Component Analysis based Time Series Segmentation – A New Sensor Fusion Algorithm. 23". In : (2004).

[2] Ionescu Alin. "Forecasting solar radiation using a deep long short-term memory artificial neural network". In : (mai 2021).

[3] Baeldung. "multi-class-f1-score". In : (19 oct. 2020). url : https://www.baeldung.com/cs/multi-class-f1-score.

[4] CAPGEMINI. "ANAGREEN : VALORISER L'ÉNERGIE PERDUE". In : (). url : https://capgemini-engineering.com/fr/fr/case_study/anagreen-valoriser-lenergie-perdue/.

[5] "Consommation énergétique secteur industrie". In : (2017). url : https://www.gazprom-energy.fr/gazmagazine/2017/09/consommation-energetique-secteur-activite/.

[6] EDF. "Production d'énergie locale : les enjeux de demain". In : ().

[7] "ENERGIES EN FRANCE : QUI CONSOMME QUOI ?" Juill. 2020. url : https://observatoire-electricite.fr/IMG/pdf/oie_-_fiche_pedago_conso_energie_2020.pdf.

[8] "EUMETSTAT". In : (). url : https://www.eumetsat.int/.

[9] Bixuan Gao. "Predicting day-ahead solar irradiance through gated recurrent unit using weather forecasting data". In : (août 2019).

[10] David Hallac. "Toeplitz Inverse Covariance-Based Clustering of Multivariate Time Series Data". In : (10 juin 2017). url : https://arxiv.org/abs/1706.03161.

[11] Huaiguang Jiang. "Solar Irradiance Capturing in Cloudy Sky Days–A Convolutional Neural Network Based Image Regression Approach". In : (jan. 2020).

[12] Krzanowski. "Between-Groups Comparison of Principal Components. J. Am. Stat. Assoc. 74, 703–707". In : (1979).

[13] Pratima Kumari. "Long short term memory–convolutional neural network based deep hybrid approach for solar irradiance forecasting". In : (1er août 2021).

[14] F. LECOQ. "Intégration énérgétique de Procédés industriels". In : (2007). url : http://www.isilf.be/Articles/ISILF07p19gramme.pdf.

[15] B. Linnhoff et E. Hindmarsh. "The pinch design method for heat exchanger networks". en. In : *Chemical Engineering Science* 38.5 (1983). Number : 5, p. 745-763. issn : 00092509. doi : 10.1016/0009-2509(83)80185-7. url : https://linkinghub.elsevier.com/retrieve/pii/0009250983801857 (visité le 04/06/2020).

[16] Miodrag Lovric. "Algorithmic methods for segmentation of time series : An overview". In : (2014). url : https://www.researchgate.net/publication/317099241_Algorithmic_methods_for_segmentation_of_time_series_An_overview.

[17] *Méthode de calcul du rayonnement solaire*. Esri. url : https://pro.arcgis.com/fr/pro-app/2.7/tool-reference/spatial-analyst/how-solar-radiation-is-calculated.htm.

[18] Neethu Elizabeth Michael. "Short-Term Solar Power Predicting Model Based on Multi-Step CNN Stacked LSTM Technique". In : (15 mars 2022).

[19] Raphaël Richard. "Dérive conceptuelle". In : (). url : https://24pm.com/117-definitions/305-derive-conceptuelle.

[20] Christelle Rigollier, Mireille Lefèvre et Lucien Wald. "The method Heliosat-2 for deriving shortwave solar radiation from satellite images". In : *Solar Energy* 77.2 (2004). Number : 2 Publisher : Elsevier, p. 159-169. url : https://hal.archives-ouvertes.fr/hal-00361364.

[21] Sahar Salame. "Méthodologie de conception d'intégration énergétique". In : (30 jan. 2017). url : https://pastel.archives-ouvertes.fr/tel-01449262/document.

[22] Julien Eynard Shab Gbémou. "A Comparative Study of Machine Learning-Based Methods for Global Horizontal Irradiance Forecasting". In : (mai 2021).

[23] Singhal. “Clustering multivariate time-series data. J. Chemom. 19, 427–438”. In : (2005).

[24] “SIRTA”. In : (). URL : https://sirta.ipsl.fr/fr/home-fr-2/.

[25] Spiegel. “Pattern recognition and classification for multivariate time series. in Proceedings of the Fifth International Workshop on Knowledge Discovery from Sensor Data”. In : (2011).

[26] Stephan Spiegel. “Pattern recognition and classification for multivariate time series”. In : (août 2011). URL : https://www.researchgate.net/publication/254003630_Pattern_recognition_and_classification_for_multivariate_time_series.

[27] Stephan Spiegel et al. “Pattern recognition and classification for multivariate time series”. en. In : *Proceedings of the Fifth International Workshop on Knowledge Discovery from Sensor Data - SensorKDD '11*. San Diego, California : ACM Press, 2011, p. 34-42. ISBN : 978-1-4503-0832-8. DOI : 10.1145/2003653.2003657. URL : http://portal.acm.org/citation.cfm?doid=2003653.2003657 (visité le 07/09/2020).

[28] Stéphane Tufféry. “Distance clustering”. In : (4 avr. 2008). URL : http://data.mining.free.fr/cours/Descriptives.pdf.

[29] Aji Prasetya Wibawa. “Time-series analysis with smoothed Convolutional Neural Network”. In : (2022).

[30] Yu Xie, Manajit Sengupta et Chenxi Wang. “A Fast All-sky Radiation Model for Solar applications with Narrowband Irradiances on Tilted surfaces (FARMS-NIT) : Part II. The cloudy-sky model”. In : *Solar Energy* 188 (2019), p. 799-812. ISSN : 0038-092X. DOI : https://doi.org/10.1016/j.solener.2019.06.058. URL : https://www.sciencedirect.com/science/article/pii/S0038092X19306334.

[31] Junfei Cao Yunjun Yu. “An LSTM Short-Term Solar Irradiance Forecasting Under Complicated Weather Conditions”. In : (oct. 2019).

[32] Tingting Zhu. “Solar Radiation Prediction Based on Convolution Neural Network and Long Short-Term Memory”. In : (déc. 2021).