Lawrence Berkeley National Laboratory,
University of California
Department of Energy
1 Cyclotron Rd, Berkeley, CA 94720, USA

Université de Technologie de Compiègne, Sorbonne Universités
Département de Génie Informatique
Rue du docteur Schweitzer, 60203 Compiègne, France

Rapport de stage TN09

Développement d'une plateforme d'étude et d'exploitation pour les réseaux éléctriques intégrant du photovoltaïque et véhicules électriques

Etudiant stagiaire:
Anthony Galtier

Maître de stage:
Jonathan Coignard

Suiveur de stage UTC:
Dominique Fontaine

August 11, 2018

Remerciements

Je tiens tout d'abord à remercier mon tuteur de stage Jonathan Coignard m'avoir offert l'opportunité d'effectuer ce stage au Lawrence Berkeley National Laboratory et pour m'avoir aidé à en tirer tous les bénéfices. Je tiens également à remercier Evangelos Vrettos pour son aide et la qualité de celle-ci. Je remercie de plus Mathieu De Sahb et Pierre-Yves Garcia pour leur bonne collaboration sur les différentes mission effectuées au cours de ce stage. Je souhaite enfin remercier l'ensemble des scientifiques et membres du service administratif du GIG (Grid Integration Group) pour leur accueil et leurs contributions.

Sommaire

1 Confidentialité de certaines données

Au cours de ce stage, j'ai été amené à travailler sur des ensembles de données confidentiels. L'accord de non-divulgation en question concerne en particulier des ensembles de données représentant des modèles de réseaux de distribution d'électricité ainsi que des données de consommation sur ces réseaux. Ces données ont été mise à la disposition du projet de recherche CyDER par la société de distribution d'énergie Californienne Pacific Gas & Electric Company (PG&E). D'autres données mises à disposition par les entreprises SolarCity et ChargePoint sont aussi concernées par une clause de confidentialité. De manière à respecter ces accords, je ne peux présenter que des illustrations limitées ou floutées de certaines de mes réalisations. Je tacherais toutefois d'en présenter les enjeux, les techniques et, dans les limites imposées par ces clauses de confidentialité, d'en présenter certains résultats.

2 Résumé technique

Ce rapport revient sur les 24 semaines de stage que j'ai effectué du 5 février au 20 juillet 2018 au Lawrence Berkeley National Laboratory au sein de l'équipe de recherche du projet CyDER. Le projet CyDER consiste en le développement d'une plate-forme de co-simulation pour l'intégration de ressources d'énergies distribuées au réseau électrique et en particulier de l'énergie photovoltaïque. Après avoir présenté le laboratoire et le projet de recherche, ce rapport décrira mes contributions sur deux projets et en détaillera les réalisations techniques.

Le premier projet a consisté en du Développement Web. Il s'agissait de développer une interface utilisateur sous forme d'application web pour l'environnement de co-simulation développé par CyDER. Le second projet a été une mission de recherche impliquant de l'analyse de données et l'implémentation d'algorithmes de Machine Learning. Il s'agissait d'étudier la capacité des réseaux de distribution d'électricité à intégrer l'énergie photovoltaïque à partir de données topologiques des réseaux et de données de simulations d'installations photovoltaïques sur ceux-ci.

3 Présentation du laboratoire et du groupe de recherche

3.1 Lawrence Berkeley National Laboratory

Le Lawrence Berkeley National Laboratory (LBNL) est un laboratoire de recherche prestigieux dépendant du Département des Énergies (DoE) du gouvernement américain et géré par l'Université de Californie (UC). Le LBNL domine le campus de l'Université de Californie Berkeley (UCB). Plus de 3 200 ingénieurs et chercheurs scientifiques, dont plusieurs récipiendaires de prix Nobel y travaillent pour amener des solutions scientifiques et technologiques aux défis énergétiques les plus pressants. Les 20 différentes divisions du LBNL effectuent aujourd'hui des recherches dans les secteurs suivants:

- Bio sciences
- Informatique
- Sciences de la Terre et de l'Environnement
- Sciences des Énergies
- Technologies Énergétiques
- Sciences Physiques

Le laboratoire est un acteur économique conséquent. Son impact économique est estimé à 1.6 milliard de dollars par an avec plus de 12 000 créations d'emplois au niveau national directement liés au laboratoire[1]. Les revenus et économies générées par les découvertes du laboratoire se mesurent quant à elles en milliards de dollars.

3.2 Grid Integration Group

Le Groupe d'Intégration au Réseau (GIG, "Grid Integration Group") est un groupe de recherche de la division de Stockage d'Énergie et Ressources Distribuées du secteur des Technologies Énergétiques au LBNL. Plus de 50 chercheurs scientifiques, ingénieurs et étudiants de ce groupe effectuent des recherches dans les domaines suivants:

[1] http://newscenter.lbl.gov/2010/04/14/berkeley-lab-economic-impact

- Réseaux de distribution
- Micro-réseaux et Ressources d'Énergies Distribuées
- Simulations véhicules-réseaux (V2G, "Vehcule to Grid")
- Réponse à la demande (DR, "Demand Response")

Les projets du GIG contribuent à la transformation des réseaux électriques vers un avenir de développement durable et d'énergie propre.

3.3 Le projet de recherche CyDER

Le projet de recherche CyDER est un projet du GIG soutenu et financé par le Département des Énergies du gouvernement américain à hauteur de 4 millions de dollars sur 3 ans. La société distributrice d'énergie Californienne Pacific Gas & Electric Company (PG&E), ainsi que les entreprises SolarCity et ChargePoint sont partenaires de ce projet.

Le projet CyDER s'inscrit dans une démarche d'intégration des ressources d'énergies distribuées, en particulier des énergies renouvelables comme le solaire, aux réseaux de transmission et distribution d'électricité. Le réseau électrique Californien, comme ailleurs, a été conçu dans un contexte de production d'électricité centralisée. Toutefois, les ressources d'énergie renouvelable, comme l'énergie photovoltaïque, sont souvent exploitées de manière décentralisée sur les réseaux. La décentralisation, en particulier dans le cas de l'énergie solaire, a tendance à créer des problèmes de tension sur les réseaux de distribution. L'état des réseaux devient par ailleurs plus sensible aux pics de consommations et aux fluctuations de production de ces ressources, révélant de nouvelles problématiques de stockage d'énergie et de réponse à la demande. La popularisation des véhicules électriques et des batteries particulières sont de plus partie prenante de ces problématiques. Ces problématiques tendent à dépasser les modèles de simulations couramment utilisé pour estimer les effets d'une installation électrique sur un réseau. L'intégration des ressources d'énergies distribuées a donc créé un besoin pour de nouveaux modèles plus complets et des environnements de simulations plus complexes, permettant d'aider à la décision, d'anticiper et d'optimiser les conséquences de nouvelles installations sur les réseaux électriques. Le projet CyDER répond à ce besoin.

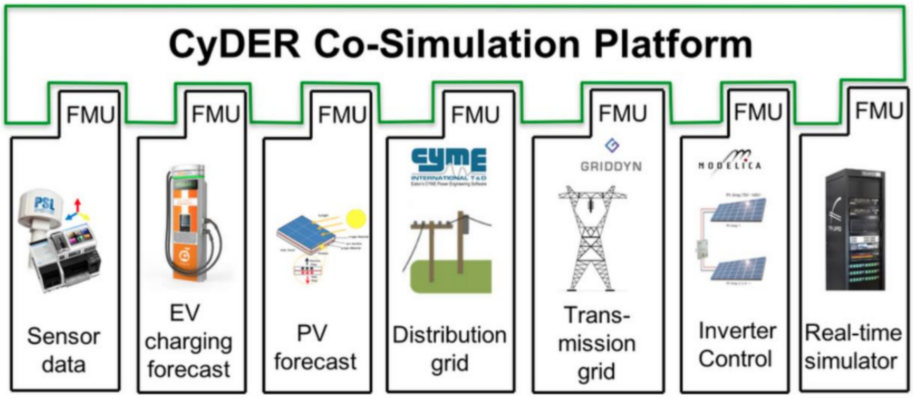


Figure 1: L'environnement de co-simuation CyDER en standard FMI

CyDER signifie "Cyber Physical Co-simulation Platform for Distributed Energy Resources (CyDER) in Smart Grids". L'objectif de CyDER est de développer un environnement de simulation, efficace et complet, en agrégeant de manière modulable et standardisée un ensemble divers d'outils de simulation pré-existants. Cet environnement combinerait des outils informatiques comme le logiciel Cyme de Eaton Corporation avec des outils physiques, hardware, comme un simulateur "Hardware In the Loop" (HIL) en temps réel.

Le standard utilisé est celui du "Functional Mock-up Interface" (FMI). Ce standard permet d'effectuer des co-simulation, encapsulant chaque module de simulation dans un "Functional Mock-up Unit" (FMU). Les différents FMUs sont ensuite mis en relations par un "master-algorithm". L'interface utilisateur de cette plate-forme de co-simulation consiste en une plate-forme web facile d'utilisation et rapide, s'adaptant aux différents cas d'utilisation, comme la validation d'une installation solaire ou d'un point de chargement de véhicule électrique sur le réseau.

A cet environnement de co-simulation s'ajoute une étude de la capacité des réseaux de distribution à intégrer des ressources d'énergies distribuées et en particulier de la capacité de ceux-ci à intégrer la production d'énergie solaire. Une partie de cette étude a été d'étudier l'importance des caractéristiques topologiques d'un réseau de distribution dans sa capacité à intégrer l'énergie solaire.

4 Présentation de la mission

Le sujet de mon stage au LBNL au sein de l'équipe de recherche du projet CyDER était le "Développement d'une plate-forme d'étude et d'exploitation pour les réseaux électriques intégrant du photovoltaïque et véhicules électriques". Ce sujet a consisté en la réalisation de deux missions, une mission de développement web et une autre d'analyse de données et Machine Learning.

4.1 Développement Web

4.1.1 Une interface utilisateur sous forme d'application web

L'interface utilisateur de l'environnement de co-simulation développé dans le projet CyDER consiste en une interface web. Celle-ci se veut facile d'utilisation, permettant de créer, gérer et exploiter les résultats de projets de co-simulation. L'application web doit permettre à un utilisateur d'exploiter aisément les différents modules de co-simulation disponibles (les différents FMUs) et de visualiser des résultats pertinents pour les différents cas d'utilisations identifiés. En ce qui concerne le développement de l'interface web, celui-ci se doit d'être modulable et extensible afin de s'adapter à tout module de simulation (FMU) et cas d'utilisation.

4.1.2 Contributions

Ma contribution a été de poursuivre le développement de cette application web en développant de nouvelles fonctionnalités et en améliorant les fonctionnalités déjà existantes. Certaines améliorations et fonctionnalités m'ont été imposées, toutefois leur implémentation ainsi que le développement de certaines autres fonctionnalités ont été de ma propre initiative.

A mon arrivée, l'application web présentait deux fonctions principales. La première permet de visualiser les différents modèles et informations associées de plusieurs réseaux de distribution d'électricité, mis à la disposition du projet CyDER par le fournisseur d'énergie PG&E. Ce visualiseur de modèles de réseaux de distribution permet d'afficher sur une carte géographique les différents nœuds et sections le composant ainsi que la répartition des charges de consommation de manière semblable à une carte thermique. La contribution que j'ai apportée a ce visualiseur a été d'améliorer son temps de chargement en modifiant les structures de données utilisées et de permettre

la visualisation spécifique des différentes lignes de chaque réseau de distribution.

L'essentiel de ma contribution en développement web a toutefois concerné la seconde fonction de l'application web, plus complexe, consistant en un gestionnaire de projet de co-simulation. Celui-ci permet, de configurer les modèles de réseaux de distribution en y imposant différentes charges de manière interactive. L'interface exploite ensuite l'API d'un module de simulation installé sur un ordinateur distant pour lancer une simulation et en récupérer les résultats. L'application web possède enfin une fonction de visualisation de résultats de simulation. Une contribution a été de revoir en profondeur les fonctions de configuration de modèles en ajoutant des fonctionnalités permettant de simuler des installations photovoltaïques et de véhicules électriques sur les modèles réseaux. Une autre contribution, en conséquence, a été de développer à neuf une fonction de visualisation de données de simulation par carte choroplèthes.

4.1.3 Outils et technologies

L'interface utilisateur est une application web développée avec le Framework Django en langage Python. L'application web exploite en back-end une base de donnée relationnelle PosgreSQL et un serveur distant responsable d'effectuer les simulations appelé "worker". La communication avec le worker est asynchrone et s'effectue avec la technologie Celery. Celery exploite une base de données Redis comme messager et s'implémente aussi dans le langage Python. L'architecture de l'application web sera explicité plus en détail dans la partie dédiée aux réalisations techniques. Le front-end consiste en des pages HTML et CSS au contenu dynamique implémenté en JavaScript. Le Framework JavaScript VueJS a été grandement utilisé dans ce contexte.

4.1.4 Prise de recul

L'interface web, bien que fonctionnelle, n'est pas un produit fini, au contraire, elle est destinée à évoluer en même temps que le projet de recherche. Le développement de celle-ci a été réalisé en accordant une importance particulière à la modularité et l'extensibilité de celle-ci. En effet, l'interface web doit être fonctionnelle avec les outils de simulation actuels, mais doit aussi permettre facilement l'intégration des nouveaux outils de simulations et source d'informations développés dans le futur de ce projet de recherche.

Au moment présent, suite à mon travail, l'interface web est fonctionnelle et pourra facilement être prise en main par un nouveau développeur. Les développements futurs de l'interface web consisteront sans doute en l'intégration de nouveaux outils de simulations et visualisation de données et probablement d'une certaine revue de la procédure de création de projet.

L'espoir porté par ces travaux est d'offrir un outil complet, facile d'utilisation et de source libre, permettant à la fois aux sociétés distributrices d'énergie, mais aussi à toute entreprise ou particulier envisageant un projet engageant des ressources d'énergie distribuées, d'évaluer la faisabilité et de mesurer les conséquences de ses installations électriques, en particulier pour le photovoltaïque, les batteries et les véhicules électriques.

4.2 Analyse de Données et Machine Learning

4.2.1 Données et problématique

La société de distribution d'énergie PG&E a mis à la disposition du projet CyDER des modélisations informatiques de 53 réseaux de distribution d'électricité en Californie du Nord ainsi que des données puissance active et réactive consommée dans le temps sur ces différents réseaux de distribution. Le projet CyDER dispose de plus d'une licence pour le logiciel Cyme permettant d'exploiter ces modèles et données de consommation. Ce logiciel dispose d'une interface graphique ainsi que d'une API en python permettant d'effectuer des simulations et analyse de réseaux électriques. A notre disposition nous avions donc, à partir des modèles, des informations sur la topologie des réseaux de distribution et d'autre part, avec le logiciel Cyme, nous avions les outils permettant de connaître les performances de ces réseaux de distribution en fonction des installations électriques qu'on y impose.

Dans ce contexte, nous nous sommes donc proposé d'étudier la capacité des réseaux de distributions à intégrer les installations solaires. Nous sommes en particulier posé les problématiques suivantes. Peut-on augmenter les installations solaires sur les réseaux de distribution d'électricité sans rencontrer de limite? Quelles seraient ces limites? La topologie d'un réseau de distribution a-t-elle une influence sur sa capacité à intégrer les installations solaires?

4.2.2 Contributions

Mes contributions dans cette mission se sont déroulées en plusieurs étapes:

La première étape consiste en la collecte des données. D'une part, il a fallu identifier et extraire les données topologiques d'intérêt des modèles de réseaux de distribution. D'autre part, il a fallu définir et développer un outil de simulation produisant des résultats de performances pertinents de différentes installations solaires.

La seconde étape consiste en l'exploration et le traitement de ces données. Les données de simulation et données topologiques ont tout d'abord été explorées et visualisées utilisant des méthodes statistiques courantes, d'abord individuellement puis confrontées les unes aux autres.

Une fois les données traitées, la troisième étape a consisté en l'application de techniques de Machine Learning, en particulier d'algorithmes de classification et de groupement ("clustering") afin d'identifier les caractéristiques topologiques discriminantes des différents réseaux de distribution ainsi que pour identifier les facteurs topologiques importants et l'influence de ceux-ci sur la capacité d'un réseau de distribution à accueillir des installations solaires.

4.2.3 Outils et technologies

L'ensemble des tâches de cette mission d'analyse de données et Machine Learning ont été effectuées avec le langage python. Les modèles de réseaux de distribution étant dans un format privé spécifique au logiciel Cyme, la collecte des données topologiques tout comme l'outil de simulation se sont appuyés sur l'API en python de ce logiciel. Ces outils ont été développés sous la forme d'un ensemble de fichiers Python et Jupyter Notebook[2]. Les bibliothèques Pandas et Numpy ont été utilisées abondamment pour manipuler et traiter les données, les bibliothèques Matplotlib et Seaborn pour la visualisation de données et enfin les algorithmes outils de "preprocessing" de la bibliothèque SciKit-Learn pour le Machine Learning.

4.2.4 Prise de recul

Cette mission d'Analyse de Données et Machine Learning fut une expérience de recherche scientifique enrichissante. L'ensemble des outils de collecte et d'analyse de données ont été développés et ont produit des résultats significatifs pour notre problématique. Nos travaux sur cette problématique et nos résultats seront le contenu d'une publication scientifique. Nous souhaiterions

[2]Plus d'informations sur le projet Jupyter sont disponibles sur le site web jupyter.org

en effet, suite à ce stage, rédiger un article de revue scientifique autour de notre problématique et des résultats que nous avons obtenu.

5 Réalisations

Les missions de Développement Web et d'Analyse de Données on été réalisées distinctement l'une de l'autre. La mission de Développement Web a couvert les 10 premières semaines de stage et la mission d'Analyse de Données et Machine Learning les 14 dernières. Dans cette partie, j'aborderais mes réalisations en conservant un certain ordre chronologique tout en détaillant les points techniques.

Je commencerai donc par la mission de Développement Web, en présentant tout d'abord son architecture et ma prise en main de celle-ci. Je détaillerai ensuite les incrémentations que j'y ai apporté, tout d'abord sur la configuration des projet de simulation et ensuite sur la visualisation des résultats.

Dans un second temps, j'aborderai la mission d'Analyse de Données et Machine Learning. Je ferai une présentation détaillée des données et du contexte qui ont donné lieu à la problématique que nous nous sommes posé. J'expliquerai ensuite la méthodologie et les outils de collecte et préparation des données en distinguant d'une part la collecte des données topologiques des réseaux de distribution et de l'autre, celles de la performance des installations solaires sur ces mêmes réseaux de distribution d'électricité. Pour chaque, je présenterai quelques analyses et visualisations de ces données. Je détaillerai ensuite les objectifs et les techniques de Machine Learning utilisées sur ces données pour répondre à notre problématique.

5.1 Développement Web

5.1.1 Architecture et prise en main

Afin d'apporter des améliorations et nouvelles fonctionnalités a cette application web, il a fallu que je prenne connaissance en détail de l'architecture de celle-ci et de l'organisation du projet informatique. L'ensemble des fichiers composant le projet informatique de l'application web est contenu dans un répertoire suivi avec le gestionnaire de version Git. Le dépôt officiel du projet est open source et disponible en ligne [3]. Des conteneurs Docker sont utilisés

[3] https://github.com/LBNL-ETA/CyDER

pour offrir un environnement de développement portable. L'application web a été développée avec le Framework Django. L'ensemble des fichiers du projet Django est contenu avec le serveur WSGI-HTTP, Gunicorn, dans un premier conteneur docker. La base de donnée PostgreSQL dispose de son propre conteneur et les échanges avec celles-ci se réalisent par le biais d'une API REST. Le serveur HTTP NGINX dispose aussi de son propre conteneur Docker. La communication avec le "worker" repose sur la technologie Celery et une base de donnée Redis qui dispose aussi de son propre conteneur Docker. La figure 2 représente schématiquement l'architecture générale de cette interface web.

N'ayant que peu de connaissance et d'expérience de ces technologies au moment de prendre en main ce projet informatique, au cours des premières de ce stage, je me suis documenté et ai effectué plusieurs tutoriels et exercices pratiques en autonomie. Je me suis en particulier formé sur la Framework Django, sur les API REST, l'usage de VueJS ainsi que sur la technologie de communication asynchrone Celery. Je me suis ensuite lancé dans de premiers développement. Mes premiers pas de développement ont concerné les pages web du gestionnaire de projet. Ces premiers pas ont consisté en une amélioration du code JavaScript en utilisant des objets et méthodes de la bibliothèques VueJS plutôt que les objets et méthodes définis sur-mesure par le développeur précédent. Utiliser les objets et méthodes ainsi que le standard de développement d'un Framework JavaScript tel que VueJS, disposant d'une excellente documentation et d'une communauté d'utilisateur importante, facilite la prise en main et les futurs développements de l'interface web.

5.1.2 Configuration des projets de simulations

La première fonctionnalité nouvelle que j'ai développée pour cette interface utilisateur a concerné les pages de configuration et d'édition de projet de simulation. La fonctionnalité est la suivante: l'utilisateur ayant sélectionné un modèle de réseau de distribution pour son projet de simulation est invité à y définir un profil de charge de consommation ainsi qu'un profil d'installations photovoltaïque. Cette fonctionnalité a été implémentée par le biais d'une carte interactive. Une carte, similaire à celle du visionneur de modèle, affiche le modèle de réseau de distribution, l'utilisateur sélectionne s'il veut y ajouter des installations photovoltaïques ou des charges de consommation avec un bouton et il peut ensuite cliquer sur n'importe quel nœud du réseau sur la carte. Un pop-up s'affiche alors au niveau de ce nœud et permet à

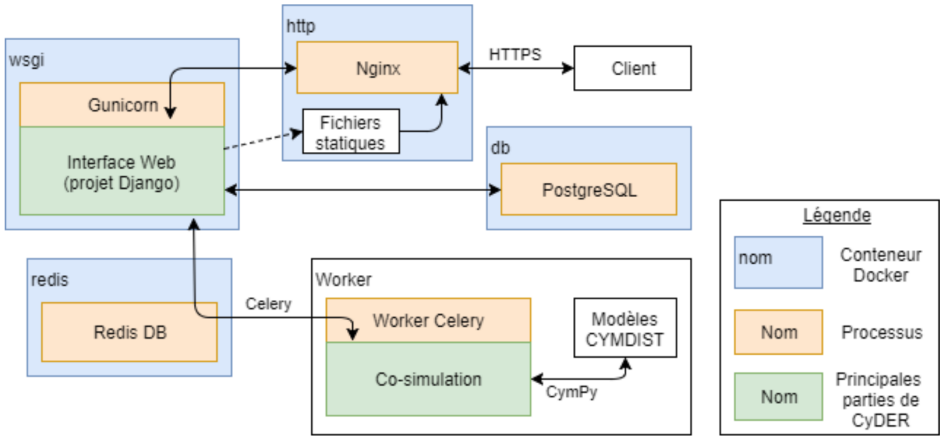


Figure 2: Architecture de l'interface web

l'utilisateur d'entrer la valeur en kilowatt (kW) de la charge de consommation ou taille de l'installation photovoltaïque. Le pop-up s'adapte au cas de manière à ce que l'utilisateur puisse aussi modifier une valeur ou supprimer une valeur précédemment entrée. Toute édition de l'utilisateur est identifiée en coloriant dynamiquement le nœud du modèle concerné. La couleur d'origine est rétablie dès que l'utilisateur supprime la valeur au nœud considéré. Les valeurs sont sauvegardées dans la base de donnée SQL lors du clic de l'utilisateur sur le bouton de sauvegarde dédié.

Pour implémenter cette fonctionnalité, dans la continuité de ce qui avait été développé pour le visionneur de modèle, je me suis appuyé sur la bibliothèque de cartographie interactive Leaflet en JavaScript couplé avec le service de source libre OpenStreetMap pour le fond de carte. Ce choix a été fait par le précédent développeur, car Leaflet et OpenStreetMap sont Open-Source et gratuits.

L'ensemble des fonctions dynamiques de cette page d'édition est implémenté grâce au Framework VueJS. Les différents éléments (élément HTML, objet ou attribut de la carte Leaflet) concernés par des fonctions dynamiques sont suivis par des objets VueJS appelés "watchers". Toute action sur ceux-ci déclenche un événement JavaScript appelant une fonction mettant à jour l'élément et rafraîchissant en temps réel son apparence graphique sur l'interface web.

La seconde fonctionnalité que j'ai développée pour cette partie de l'interface web responsable de la création et configuration de projet de simulation a con-

cerné essentiellement le back-end. En effet, suite à une discussion avec mon tuteur de stage, au lieu de demander à l'utilisateur de déclarer soi-même les dates et heures sur lesquelles il souhaite effectuer sa simulation, nous avons décidé de les sélectionner basé sur un calcul de charge nette minimale. Plus la charge nette est basse, plus un réseau de distribution sera susceptible de présenter des problèmes de voltage. Dans notre cas, la charge nette minimale sur un réseau de distribution électrique est définie comme la différence entre la charge de consommation totale et la génération d'électricité totale des installations photovoltaïques en fonction du temps. A partir d'un ensemble de données datées d'irradiation solaire en Californie du Nord et des ensemble de données datées de consommation sur les différents réseaux de distribution fournis par PG&E, j'ai implémenté ce calcul en python et l'ai intégré à l'interface web de manière à ce que ce calcul soit effectué lors de l'import des modèles depuis le worker. La gestion des ensembles de données et les calculs reposent sur les bibliothèques Pandas (DataFrame et Series) et Numpy ainsi que des objets python développés par mon maître de stage permettant d'estimer la génération d'électricité photovoltaïque dans le temps d'une installation de puissance donnée en fonction des données d'irradiation solaire.

5.1.3 Visualisation des résultats de simulation

L'interface web doit permettre à l'utilisateur de visualiser les résultats de simulation pertinents pour évaluer la faisabilité et la performance de ses installations. La première étape en ce qui concerne les fonctionnalités de visualisation de ces résultats à été de revoir la procédure de lancement et rapatriement des résultats de simulation. Mon tuteur de stage a en effet développé un module en python exploitant l'API du logiciel de simulation Cyme retournant de nouveaux résultats plus complet sous forme de DataFrame pandas contenant notamment les résultats de voltage par phase à tous les nœuds du modèle de réseau de distribution. J'ai donc intégré ce module sous la forme de tâches Celery. Ces tâches sont en fait des objets en python définis dans des fichiers python spécifiques régissant la communication asynchrone entre le worker et l'interface web avec la technologie Celery. Cette communication asynchrone implémenté avec la technologie Celery repose sur une base de données noSQL Redis qui retient l'identifiant et l'ordre des tâches appelées par l'interface web. Ces tâches sont exécutées dans l'ordre d'appel du côté du worker et les valeurs retournées sont sauvegardée dans la base de données

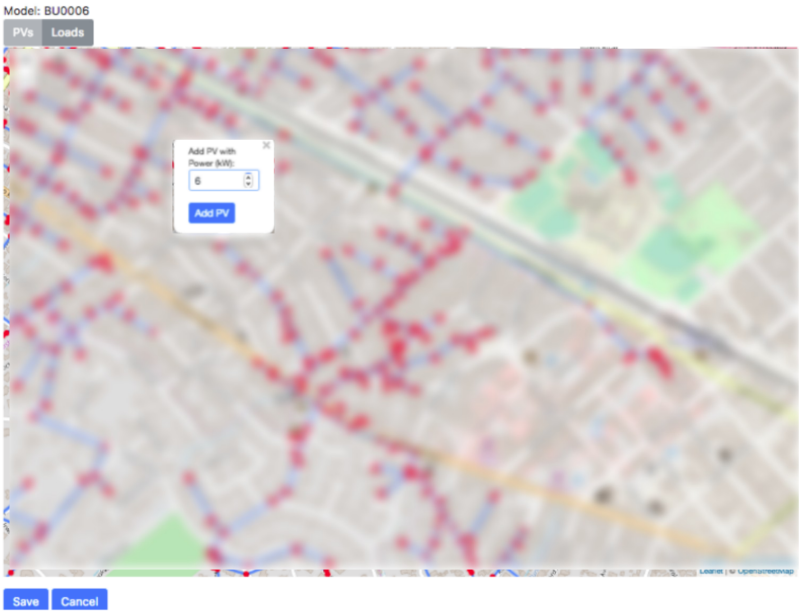


Figure 3: Carte interactive de configuration des projets de simulation

Redis. Un attribut de la tâche indique alors que la tâche est réalisée avec succès et à ce moment la, les résultats sont rapatriés vers la base de données PostgreSQL de l'interface web par des requêtes sur l'API REST. Un point clé de cette réalisation a été l'optimisation des structures des données dans les modules de simulation et rapatriement de données de manière à ce que les simulations s'effectue dans des temps raisonnables. Cette optimisation a essentiellement consisté en la préférence de dictionnaire plutôt que de liste ou tableaux lorsque possible et la minimisation de l'usage de pandas DataFrame sur lesquels les opérations de copie et d'écriture sont assez peu performantes. J'ai réussi a maintenir la complexité de ces tâches en O(2), permettant ainsi de réaliser et collecter les résultats d'un projet de simulation en moins d'une minute, soit 80% plus rapidement qu'auparavant.

Une fois la procédure de lancement et rapatriement des résultats de simulation mise en place, j'ai été libre de trouver une manière pertinente de présenter graphiquement les résultats à l'utilisateur. Pour cette page web de visualisation des résultats de simulation, j'ai choisi de réaliser une carte choroplèthe interactive. L'idée de cette carte est de colorer les nœuds et branches du modèle selon une échelle couleur en fonction de leur valeur de

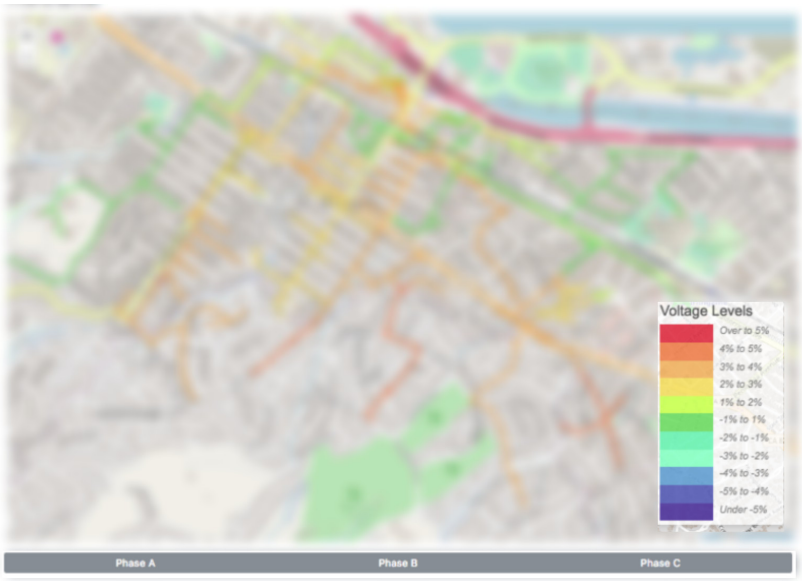


Figure 4: Carte choroplèthe des résultats de simulation

voltage mesurée lors des différentes simulations. En cliquant sur un nœud ou branche du modèle, un pop-up affiche la valeur exacte de voltage mesuré en ce point. Cette carte est complétée par deux graphes affichés en bas de page. Un premier graphe en nuage de points représente le voltage des différents nœuds en fonction de leur distance de la tête du réseau de distribution. Le second graphe représente le profil de charge et la génération d'électricité des installations solaires en fonction du temps. Les résultats affichés par la carte et ce premier graphe varient selon la date et heure de simulation sélectionnée par l'utilisateur dans un menu déroulant en haut de la page. Dès que l'utilisateur sélectionne une nouvelle date, la carte et le premier graphe se mettent à jour. Le second graphe quant à lui et statique et doit permettre justement à l'utilisateur de sélectionner les dates et heures de simulation qui lui sont d'intérêt. L'implémentation a été réalisée à nouveau avec un fond de carte OpenStreetMap et la bibliothèque Leaflet. Les fonctions dynamiques, tout comme pour la page d'édition sont implémentées avec le Framework VueJS en Javascript. J'ai utilisé la bibliothèque Plotly en Javascript pour réaliser les graphes. La difficulté principale rencontrée dans ce développement à été d'assurer un temps de chargement rapide de la page et une réaction rapide des éléments graphiques dynamiques. Il a fallu à

nouveau prendre soin au moment de chargement de la page de lire et organiser les données dans des structures de données rapides d'accès spécifiques à notre cas. J'ai réussi à réduire le temps de chargement à moins de 4 secondes contre plus de 30 secondes avant optimisation. Au chargement de la page, les résultats de voltage des différents nœuds et branches sont ajoutés sous forme d'attributs aux données géographiques du modèle en GeoJSON permettant à la carte d'être très réactive lorsque l'utilisateur change de date et heure de simulation dans le menu déroulant. La carte et le graphe sont mis à jour moins d'une demi-seconde après l'action de l'utilisateur.

5.1.4 Prise de recul

Ces réalisations sur l'interface web ont permis d'ajouter une certaine valeur au projet de recherche CyDER. Les fonctionnalités implémentées par celle-ci ont en effet été d'un intérêt particulier pour la société de distribution d'énergie PG&E. Cet outil sous forme d'interface web pourrait en effet être d'une certaine utilité aux ingénieurs pour rapidement et facilement évaluer la faisabilité et les performances de leurs projets d'installations électriques et identifier des sections défaillantes nécessitant des travaux pour répondre à la demande croissante d'installations photovoltaïques, de véhicules électriques et batteries particulières.

D'un point de vue personnel, je suis satisfait de mes travaux sur cette application web. J'ai su prendre en main et avancer en autonomie sur un projet d'application web d'architecture relativement complexe et mettant en jeu plusieurs technologies nouvelles. J'ai pu acquérir de bonnes compétences en développement web, en particulier avec le Framework Django, les APIs REST, que je saurais valoriser dans mes futurs projets.

5.2 Analyse de Données et Machine Learning

La mission d'analyse de données et machine learning a débuté à la mi-avril, une fois la mission de développement web terminée et a été mon sujet principal de travail jusqu'à la fin de ce stage. Je souhaitais au cours de ce stage trouver l'opportunité d'acquérir de l'expérience en science de données. Mathieu De Sahb, l'autre étudiant stagiaire de l'UTC également en stage sur le projet CyDER partageait cet intérêt. Plusieurs ensembles de données ayant été mis à la disposition du projet de recherche CyDER, nous avons réfléchi avec Jonathan Coignard, mon tuteur de stage et défini une problématique à étudier en exploitant ces données. Mathieu et moi avons donc réalisé cette étude en collaboration, Mathieu investissant toutefois seulement un tiers de son temps à ce projet.

5.2.1 Ressources et Problématique

L'étude que nous avons mené repose sur d'une part sur des données directement mises à la disposition du projet CyDER sous forme fichier CSV ou JSON. Toutefois, l'essentiel des données utiles à notre étude n'ont pas été directement disponibles. Il a fallu développer plusieurs outils en python pour rapatrier et produire les données pertinentes pour notre étude en exploitant différents outils logiciels à notre disposition.

Les données les plus essentielles à notre étude ont été fournies au projet CyDER par la société de distribution d'énergie PG&E. PG&E a partagé avec CyDER 53 modèles informatiques de réseaux de distribution d'électricité en Californie du Nord. Chaque modèle consiste en un fichier au format .sxst, format privé exploitable uniquement avec le logiciel Cyme de EATON pour lequel le projet CyDER dispose d'une licence. PG&E a aussi contribué des ensembles de données la puissance active en kilowatt (kW) et réactive en kilovar (kVar) consommée au cours de l'année 2016 sur ces différents réseaux de distribution. A chaque modèle correspond un fichier au format CSV dont chaque paire de colonnes correspond à la puissance active et réactive consommée sur une branche donnée du réseau de distribution d'électricité. Les lignes sont indexées par la date et heure des mesures correspondantes. Il y a une mesure de consommation par heure pour les 12 mois de l'année 2016 pour chaque modèle sauf deux dont les mesures sont pour 2017.

Toutes les ressources, modèles et données, partagées par PG&E dans le cadre du projet CyDER sont confidentielles. Un accord de non-divulgation a

été signé par les dirigeant du projet de recherche. Cet accord nous contraint en particulier à garder anonyme la localisation géographique spécifique des différents réseaux de distribution de manière, entre autres, à s'assurer que la consommation de leurs clients reste anonyme.

En complément à ces ressources de PG&E, nous nous sommes aussi procuré un ensemble de données mesurant l'irradiation solaire au niveau d'une station à proximité des réseaux de distribution d'électricité dont nous avions les modèles. Ces données d'irradiation solaire ont été tirées de la base de donnée "National Solar Radiation Data Base"[4] (NSRDB) du National Renewable Energy Laboratory[5] (NREL). L'ensemble de données que nous avons téléchargé consiste en un fichier CSV relevant les mesures de l'irradiation solaire GHI, "Global Horizontal Irradiance" (Rayonnement Solaire Horizontal). Ces mesures sont indexées par leur date et heure correspondante. Il y a une mesure de par heure pour les 12 mois de l'année 2016. Ces données sont libres d'accès et ne sont soumises à aucune clause de confidentialité.

Nous avions donc à partir de ces ressources les moyens d'obtenir suffisamment de données pour d'une part décrire la topologie des réseaux de distributions d'électricité. Les modèles permettent en effet d'avoir les informations sur tous les éléments constituant les réseaux de distribution ainsi que les paramétrages et organisation géographique de ceux-ci. Ces éléments composants un réseau de distribution sont par exemple les transformateurs, les lignes et si elles sont monophasées ou triphasées... Les modèles apportent aussi des informations sur les consommateurs sur ces réseaux en particulier sur leur type (consommateurs particuliers, industriels, agriculture ou autre). D'autre part, nous avions accès au logiciel Cyme qui nous permettait de soumettre ces modèles de réseau à des simulations d'installations électriques et en particulier d'installations photovoltaïques. De ces simulations, nous pouvions obtenir une estimation des niveaux de voltage en tout points du réseaux dans le temps en fonction de paramètre de consommation et d'irradiation solaire donné. Nos données et ce logiciel nous permettaient donc d'évaluer les performances des réseaux de distribution en fonction des installations électriques qu'on y imposerait.

Étant donné ce contexte et le sujet du projet de recherche CyDER nous avions suffisamment de matière pour réaliser une étude de la capacité des

[4]La base données peut être visualisée et téléchargée librement depuis l'adresse suivante: maps.nrel.gov/nsrdb-viewer/

[5]Site web du NREL: www.nrel.gov/

réseaux de distribution à intégrer les installations solaires. Nous nous sommes donné pour problématique les questions suivantes: Peut-on augmenter les installations solaires sur les réseaux de distribution d'électricité sans rencontrer de limite? Quelles seraient ces limites? La topologie d'un réseau de distribution a-t-elle une influence sur sa capacité à intégrer les installations solaires?

5.2.2 Collecte des données

Afin d'obtenir suffisamment de données pour réaliser une étude convenable de notre problématique nous avons développé un ensemble d'outils en python interagissant avec l'API du logiciel Cyme. Cette étape de collecte de données s'est organisée en deux tâches distinctes. D'une part, la collecte des données topologiques, et de l'autre, la collecte des données de simulations pour évaluer la performance des différents réseaux face aux installations photovoltaïques.

5.2.2.1 Données topologiques

L'outil de collecte de données topologiques interagit avec l'API python du logiciel Cyme pour collecter et ensuite formater les données topologiques de l'ensemble des modèles de réseaux de distributions. Il consiste tout d'abord en ensemble de fonctions python permettant pour un fichier de modèle considéré, d'effectuer des requêtes sur l'API du logiciel Cyme pour une sélection de mots-clés correspondant à l'information topologique souhaitée. Ces mots-clés sont définis et accessibles dans la documentation de l'API du logiciel Cyme. Malheureusement, cette documentation est vieillissante et assez mal organisée. L'identification des mots-clés correspondant aux informations topologiques souhaitées fut donc un travail assez fastidieux. L'optimisation de ces fonctions d'accès à l'API a été un point crucial de ce développement. Il était, en effet, nécessaire d'optimiser la complexité de ces fonctions de manière à avoir un temps d'exécution raisonnable nous permettant d'effectuer des test et corrections rapides de notre code. Un fichier python consistant en une fonction "main", itère sur l'ensemble des 53 fichiers de modèles de réseaux de distribution, et à partir d'un dictionnaire donné en entrée de la fonction par l'utilisateur, fait appel aux fonctions d'accès nécessaires pour rapatrier les données. Ces données sont ensuite organisées dans DataFrame de la bibliothèque Pandas et ensuite exportées dans 4 fichiers au format CSV. Chacun des 4 fichiers CSV contient respectivement des informations sur les "Dispositifs" présents sur les réseaux de distribution (transformateurs, bat-

teries...), sur les différents "nœuds", ou point de connexion du réseau, sur les "sections" ou lignes composant le réseau et enfin sur le réseau lui-même (sa longueur maximale, le nombre de consommateurs...). Un Notebook Jupyter permet ensuite de lire ces 4 fichiers CSV, de regrouper et agréger intelligemment les données dans un seul DataFrame Pandas. Ce DataFrame Pandas est ensuite exporté au format CSV et c'est ce fichier qui sera exploité dans les travaux de Machine Learning.

5.2.2.2 Données de performance

Cette étape a consisté en la collecte de données mesurant les performances des différents réseaux de distribution d'électricité en fonctions des installations solaire qu'on y imposait. Cette étape fut déterminante et c'est sans doute celle qui a nécessité la plus grande étude et réflexion en amont de son implémentation. Il a en effet fallu que je m'instruise sur le génie électrique des réseaux de distribution avec l'appui de mon tuteur de stage pour pouvoir définir quelles installations photovoltaïques j'allais simuler et sous quelles configurations.

L'objectif est d'étudier comment un réseau distribution se comporte lorsqu'on augmente la génération d'électricité photovoltaïque sur celui-ci. Les mesures de voltage sont déterminantes pour évaluer la performance d'un réseau de distribution face aux installations photovoltaïques qu'on y impose puisque la génération photovoltaïque a tendance à augmenter la tension des lignes et causer des problèmes de survoltage. La configuration de répartition sur le réseau de distribution des installations est aussi significative en ce qui concerne ces problèmes de surtension. Plus l'installation photovoltaïque est éloignée de la source du réseau de distribution, plus grands tendent à devenir les problèmes de surtension. Notre intérêt est de révéler les limites d'un réseau en terme d'intégration de la génération solaire. On souhaite ainsi se concentrer sur les conditions dans le pire des cas. Le temps devient alors une dimension importante de notre étude puisque les charges de consommation, tout comme la génération solaire et par conséquent la tension sur le réseau évoluent dans le temps. De ces informations j'ai donc mis au point une méthodologie et l'ai implémenté sous la forme de Jupyter Notebook en Python.

Pour une valeur n donnée par l'utilisateur, l'outil effectuera des simulations pour n^2 configurations d'installation photovoltaïque. D'une part, la capacité de génération totale des installations photovoltaïques variera de

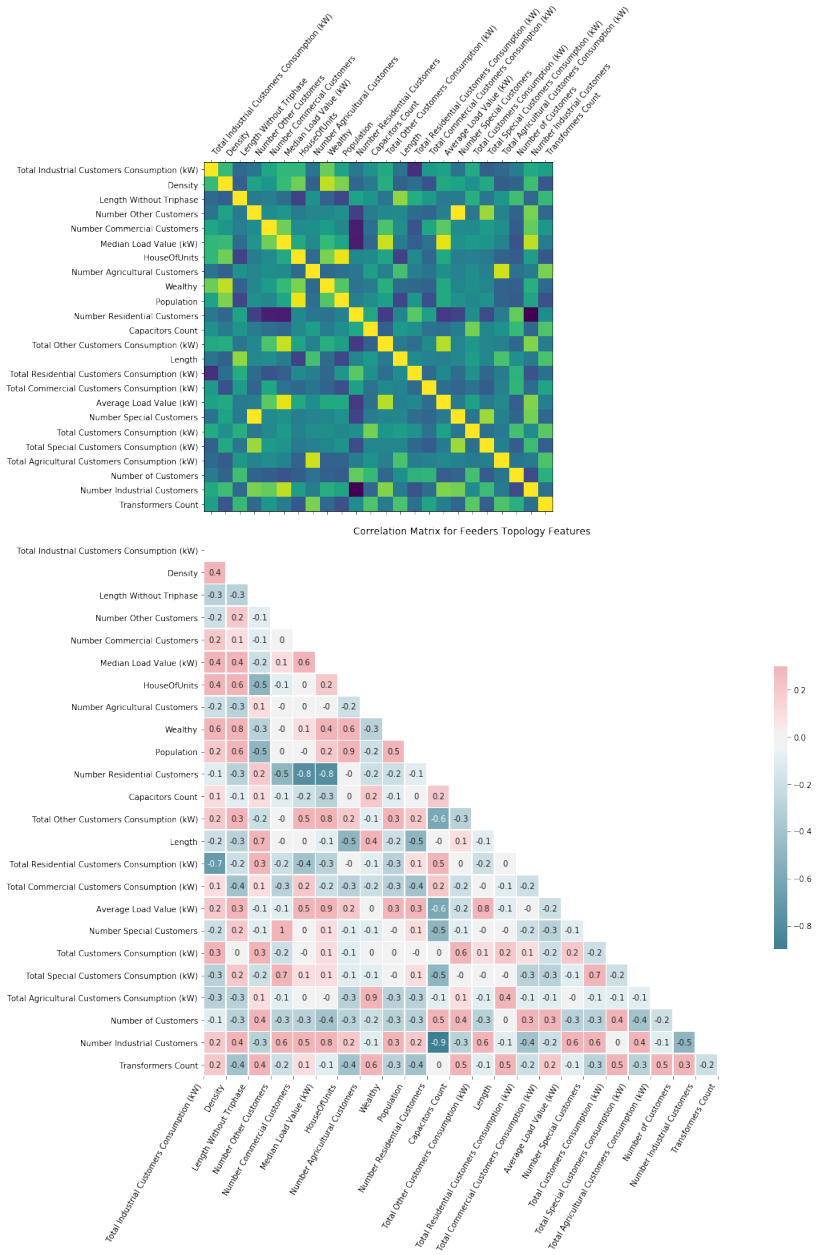


Figure 5: Corrélations entre les éléments topologiques

manière à ce que le taux de pénétration en énergie photovoltaïque du réseau de distribution varie sur n valeurs réparties uniformément entre 0% et 125%. D'autre part, la distance minimale par rapport à la source du réseau au-delà de laquelle les installations photovoltaïques sont appliquées variera sur n valeurs uniformément distribuées entre 0% et 100% de la longueur maximale du réseau de distribution. Les installations photovoltaïques sont réparties uniformément sur la portion de réseau considérée. Cette répartition uniforme s'appuie sur les nœuds du modèle de réseau de distribution. Ces nœuds sont en effets identifiés et l'API du logiciel Cyme nous permet de connaître leur distance de la source du réseau. Les nœuds dans la portion considérée sont ainsi retenus dans un dictionnaire et la fraction correspondante de la capacité de génération photovoltaïque totale leur est assignée. Ces valeurs seront passées en paramètres au lancement des simulations à travers l'API de Cyme.

L'étape essentielle et qui est implémentée en tête du Jupyter Notebook a été de donner une définition du taux de pénétration en énergie photovoltaïque d'un réseau de distribution d'électricité. Nous nous sommes appuyés sur les données d'irradiation du NREL et les données de consommation de PG&E du réseau considéré. Nous avons considéré deux méthodes. La première considère que l'on atteint 100% de taux de pénétration lorsque la puissance totale en kW des installations photovoltaïques atteint la valeur maximale de la consommation totale en kW mesurée sur ce même réseau. La seconde méthode se base sur l'énergie plutôt que la puissance. Les deux méthodes ont été implémentées en Python dans le Jupyter Notebook et c'est le résultat de la seconde méthode qui est retenu dans la suite.

Les dates et heures de simulations sont sélectionnées de manière à ce que celles-ci correspondent aux heures de charge nette minimale du réseau considéré. La charge nette sur un réseau à un temps donnée est calculée comme la différence entre la charge de consommation totale en kW et la génération photovoltaïque totale en kW. A partir des données d'irradiation, on calcule une estimation de la puissance générée à un temps donnée pour une installation photovoltaïque de 1MW (valeur choisie arbitrairement). On effectue ensuite ce calcul de la charge nette aux dates et heures correspondante avec les données de consommation de PG&E. 16 heures sont retenues pour les simulations, huit heures pour chacun des deux jours de charge nette minimale.

L'ensemble de ces données d'entrées sont retenues dans un dictionnaire passé en paramètre d'une fonction interagissant avec l'API du logiciel Cyme

pour lancer l'ensemble des simulations et retourner les résultats souhaités. Initialement, je souhaitais retourner pour chaque simulation les résultats de tension de tous les nœuds du réseau de distribution. Toutefois, pour une valeur n entrée par l'utilisateur n^2*16 simulations sont effectuées pour un seul réseau de distribution. Le nombre de nœuds d'un modèle pouvant dépasser 7000, le nombre de requêtes à effectuer sur l'API était très élevé rendant le temps d'exécution très peu performant, de l'ordre de 2 simulations par minute seulement. En fin de compte, j'ai choisi de rapatrier, pour chaque simulation, seulement les données essentielles à notre étude, c'est-à-dire:

- La tension maximale et minimale mesurée à travers l'ensemble du réseau en pu ("per unit")
- Puissance active et réactive totale mesurée à la source du réseau en kW et KVar respectivement
- La puissance générée au niveau des installations photovoltaïques en kW
- La tension mesurée au niveau des installations photovoltaïques en pu

Ces données sont retournées dans un dictionnaire de manière à optimiser le temps d'exécution des simulations. L'optimisation des structures de données utilisées dans l'ensemble du Jupyter Notebook a permis d'atteindre un temps d'exécution bien plus performant, de l'ordre de 120 simulations par minute. Les dictionnaires sont ensuite formatés dans un DataFrame pandas dont chaque ligne correspond aux résultats d'une simulation identifiée par son index et les colonnes décrivant sa configuration et sa date et heure de simulation. Ce DataFrame est enfin exporté dans un fichier CSV.

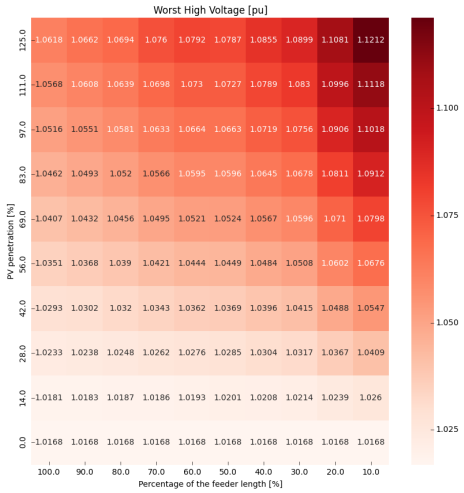


Figure 6: Tension maximale et puissance active maximale mesurées en fonction de la configuration de simulation

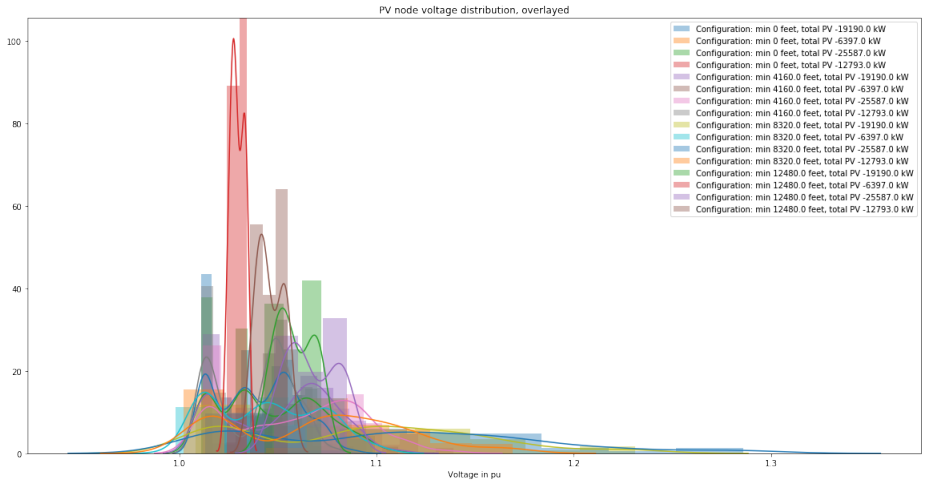


Figure 7: Distribution des nœuds d'un modèle de réseau en fonction de la tension mesurée et de la configuration de simulation

5.2.3 Machine Learning

5.2.3.1 Objectifs

A partir des des ensembles de données topologiques et de performance produits précédemment, nous souhaitions soumettre ces données à différents algorithmes de Machine Learning pour apporter des éléments de réponse à notre problématique. Nous avons d'une part soumis l'ensemble des données à des algorithmes de classification pour évaluer à quel point les facteurs topologiques déterminent sa capacité à intégrer les installations photovoltaïques. D'autre part, nous avons soumis uniquement les données topologiques à des algorithmes de groupement afin d'identifier de potentielles topologies caractéristiques des réseaux de distribution. Les résultats des algorithmes de groupement topologiques ont ensuite été confrontés aux résultats de simulations de manière à identifier si certaines topologies favorisent ou non l'intégration du photovoltaïque. Pour ces travaux de Machine Learning, nous avons exploité les outils et l'excellente documentation de la bibliothèque scikit-learn[6].

[6]http://scikit-learn.org

5.2.3.2 Algorithmes de classification

La première étape avant d'implémenter les modèles de classification a été de préparer les données. J'ai donc créé un Jupyter Notebook lisant les deux ensembles de données dans deux DataFrames Pandas et effectuant ensuite les opérations d'agrégation et de jointure permettant de fusionner les données dans un seul DataFrame pandas dont toutes les colonnes sauf la dernière correspondent à des facteurs topologiques. La dernière colonne correspond à la tension maximale mesurée sur le réseau de distribution à travers toutes le simulation d'une même configuration d'installation photovoltaïque. Chaque ligne correspond donc à une configuration de simulation (taux de pénétration, distance minimale des installations photovoltaïques) pour un réseau de distribution donné.

Cet ensemble de données est ensuite imputé de manière à pouvoir être traité par les algorithmes de classification de la bibliothèque scikit-learn. Cette imputation consiste à remplacer les valeurs nulles de l'ensemble de données par la moyenne des valeurs de la colonne. De manière à optimiser les performances des algorithmes de classification, les données sont ensuite redimensionnée entre 0 et 1. Ce redimensionnement est effectué de manière à ce que le 0 corresponde à la valeur minimale de la colonne et le 1 au maximum. Les autres valeurs sont ensuite redimensionnées proportionnellement. Nous avons choisi de redimensionner les valeurs de cette manière-ci plutôt que de les standardiser selon la loi normale, car les valeurs de la majorité des facteurs topologiques d'un réseau de distribution ne suivent justement pas une loi normale.

L'ensemble de données est ensuite séparé en un ensemble d'entraînement et un ensemble de test. L'ensemble d'entraînement contient les données de 80% des réseaux de distribution sélectionnés aléatoirement et l'ensemble de test les 20% restant. Une sélection de colonnes correspondant aux facteurs topologiques à étudier est alors utilisée pour entraîner plusieurs modèles de classification basée sur la colonne des tensions maximales mesurées et plus spécifiquement si celle-ci dépasse la limite de 1,05pu. La tension d'un réseau de distribution est, en effet, considérée comme convenable tant que celle-ci reste entre 0,95pu et 1,05pu. Une fois les modèles entraînés, leurs performances sont évaluées par leurs prédictions sur l'ensemble de données de test. Afin d'accroître la confiance en nos résultats nous avons effectué une "cross-validation" de ceux-ci. Parmi les modèles entraînés, les suivants ont donnés des résultats significatifs, prédisant correctement plus de 70% des

résultats correctement pour un certain sous ensemble de facteurs topologiques sélectionné:

- Nearest Neighbors
- Linear Support Vector Machines
- Gaussian Process
- Decision Tree
- Random Forest
- Adaptive Boost

Il a ensuite été intéressant, pour les modèles le permettant, d'étudier la variance expliquée de chacun des facteur topologique dans la construction du modèle de classification afin d'évaluer justement l'influence de ce facteur dans la classification selon les performances en terme de tension des installations photovoltaïques.

5.2.3.3 Algorithmes de groupement

Contrairement aux modèles de classification, les modèles de regroupement ne sont pas supervisés. Les différents modèles de regroupement ont ainsi été entraînés exclusivement sur les données topologiques, indépendamment des résultats de simulations. L'apprentissage n'étant pas supervisé, la performance du modèle ne peut pas être évaluée avec un ensemble de données de test comme on l'a fait avec la classification. Les modèles sont donc construits sur l'ensemble de données topologiques complet. Toutefois, pour les mêmes raisons et de la même manière qu'avec classification, l'ensemble de données est imputé avec les valeurs moyennes et redimensionné entre 0 et 1 proportionnellement au minimum et maximum pour chaque colonne, c'est-à-dire pour chaque facteur topologique.

Avant de soumettre ces données aux algorithmes de regroupement, nous avons effectué une Analyse des Composants Principaux (PCA) afin d'identifier de potentielles dépendances entre les facteurs topologiques et d'identifier les facteurs discriminants le plus les réseaux de distribution entre eux. La PCA réduit de plus la dimensionnalité de notre étude. En effet, la PCA permet de

trouver les composants principaux sous forme de combinaisons linéaires orthogonales des facteurs topologiques (des colonnes) maximisant la variance expliquée. Ce sont ainsi les valeurs de variance expliquée de chaque composant principal et les coefficients attribués à chaque facteurs qui nous permettent de comprendre l'importance de chaque facteur dans la différentiation des réseaux de distribution d'électricité. La réduction de dimensionnalité de la PCA nous a aussi permis de visualiser ces données topologiques de réseaux de distribution. En effet, les deux premiers composants principaux de l'ensemble de données permet d'expliquer déjà plus de 65% de la variance totale des facteurs topologiques et en ajoutant le troisième composant on explique presque 80% de cette variance totale. Il est donc possible de produire des représentations bidimensionnelles ou tridimensionnelles représentant relativement bien les ressemblances et différences entre les topologies des réseaux de distribution.

L'étape suivante fut d'entraîner des modèles de groupement sur cet ensemble de données topologiques. Plusieurs modèles ont été entraînés avec plusieurs paramétrages et données différentes. Pour chaque modèle entraîné, il était visuellement intéressant de reproduire les représentations graphiques bidimensionnelles ou tridimensionnelles de la PCA en colorant les points selon le groupe qui leur est attribué. En fonction des résultats que nous cherchions à révéler, il était intéressant de considérer seulement un sous-ensemble de facteurs topologiques. En faisant ceci, nous réduisions encore une fois la dimension de notre étude et ceci avait l'avantage d'améliorer l'interprétabilité des résultats et ainsi d'augmenter notre confiance en nos résultats. Les modèles de regroupement que nous avons retenu pour notre étude sont les suivants:

- K Means
- Mean Shift
- DBSCAN
- Affinity Propagation

La dernière étape a consisté en l'exploration des groupes identités par les différents modèles et d'en comparer les résultats de simulation d'installation photovoltaïque. Nous avons donc effectué une jointure entre l'ensemble de données topologiques, les résultats des algorithmes de groupement et l'ensemble de données de performances des réseaux de distribution. Certains

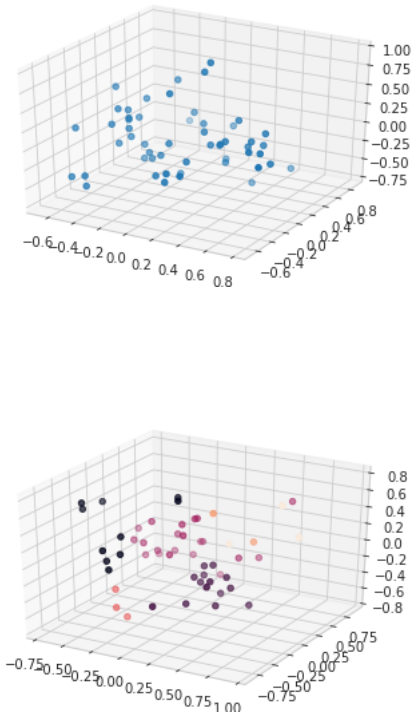
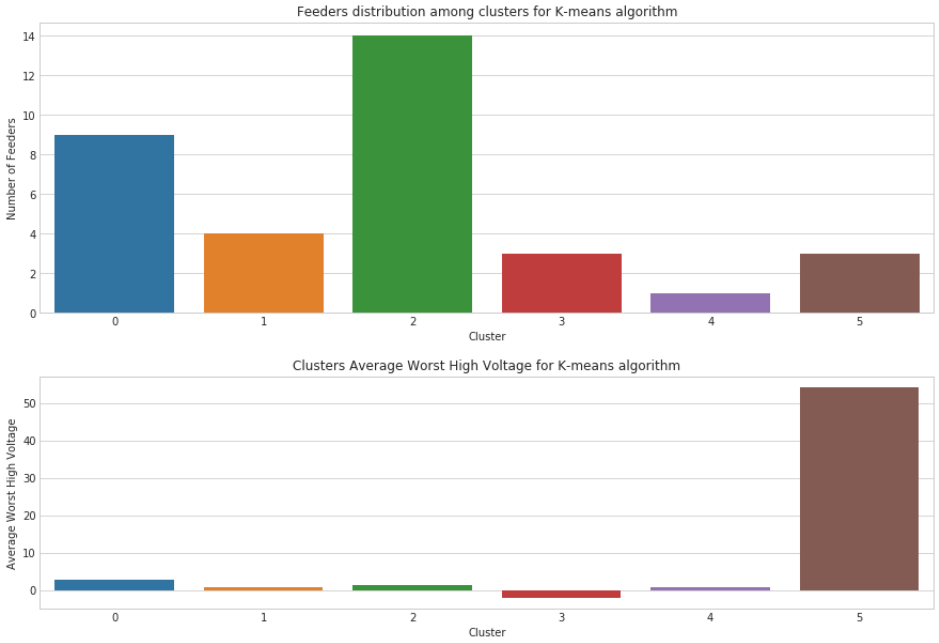


Figure 8: Distribution des modèles de réseaux selon les trois premiers Composant Principaux de la PCA et colorée (en bas) selon le groupement de l'algorithme K-Means



algorithmes groupant les topologies de réseaux de distribution en fonction d'une certaine mesure de distance comme K Means ou Mean Shift ont été ici très intéressants. On peut connaître de ces modèles les "valeurs centrales" de chaque groupe identifié. Ceci permet de faire des inférences sur les performances d'intégration du photovoltaïque de réseaux de distribution "typiques", représentés par ces "valeurs centrales" de chaque groupe. Au contraire, les modèles se basant sur des mesures de densité plutôt que de distance comme DBSCAN, bien que très efficace dans d'autres cas, ne nous ont pas été très utiles ici.

5.2.4 Prise de recul

Notre étude au moment présent, n'est pas tout à fait terminée. Les outils permettant de produire, manipuler et analyser les données sont bien en place, toutefois, nous sommes limités par la quantité de données mises à notre disposition. L'ensemble des analyses et résultats ont été produits étudiant un ensemble de 53 modèles de réseaux de distribution dont seulement 38 retournent des résultats de simulations convenables. Ce nombre de modèles mis à notre disposition est une grande limite de notre étude. Il est bien évidemment préférable que ce nombre soit plus conséquent pour convenable-

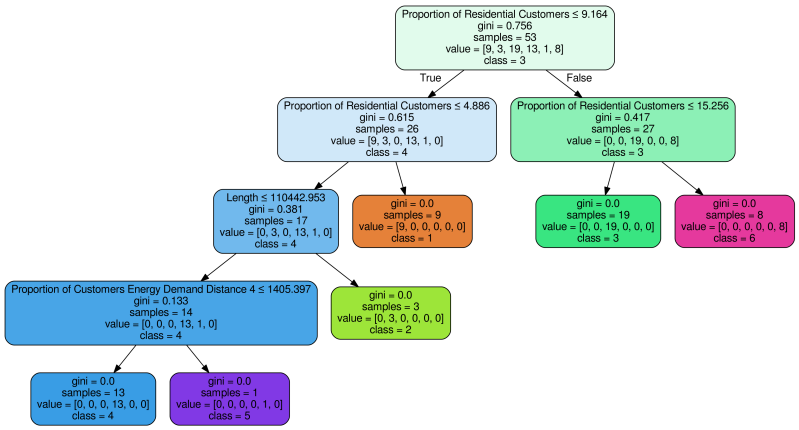


Figure 10: Règles de l'arbre de décision définissant un groupement de modèles identifié par l'algorithme K-Means

ment entraîner les modèles de classification et de groupement et avoir une plus grande confiance en nos résultats. Nous espérons voir plus de modèles de réseaux de distribution d'électricité mis à notre disposition et envisageons de rédiger une publication scientifique afin de présenter nos résultats.

6 Conclusion

Ce stage effectué au LBNL fut en tout point de vue une expérience bénéfique et très valorisable. L'expérience acquise d'une part en développant l'application web de l'environnement de co-simulation du projet CyDER m'a permis d'acquérir les compétences fondamentales du développement web. Avoir des compétences en développement web devient selon moi essentiel à un ingénieur dans ce contexte actuel où presque tous les systèmes d'informations reposent en partie sur le réseau internet. Bien que le développement de l'interface utilisateur soit une tâche importante du projet CyDER, celle-ci ne contribuait pas directement à la recherche scientifique menée.

La seconde tâche, contrairement à la première, fut une contribution directe à la recherche scientifique menée. La méthodologie développée pour notre étude de l'intégration de l'énergie photovoltaïque aux réseaux de distribution d'électricité pourra être intégrée à l'environnement de co-simulation de CyDER. Les résultats tirés de cette étude sont de plus destinés à être présentés dans une publication scientifique future du projet CyDER. Ce travail, contribuant ainsi directement à la recherche scientifique du projet, a été personnellement valorisant et m'a permis de m'intégrer plus fortement dans l'équipe. Pour cette étude de l'énergie photovoltaïque, j'ai bénéficié d'une grande liberté dans la définition de la problématique et de la méthodologie. C'est ainsi que j'ai pu m'offrir l'opportunité d'acquérir de l'expérience en analyse de données et en Machine Learning. Il me tenait particulièrement à cœur de trouver une opportunité d'effectuer de l'analyse de données lors de stage. Cette expérience m'a conforté dans mon choix de poursuivre une spécialisation en fouille de données et décisionnel dans la suite de mes études d'ingénieur.

A travers ce stage au sein de l'équipe du projet CyDER, j'ai contribué à de la recherche scientifique à la pointe du domaine des ressources d'énergies distribuées dans un laboratoire reconnu mondialement. L'environnement de travail dans lequel j'ai évolué a été enrichissant de part les rencontres que j'ai pu y faire. Le laboratoire regroupe un ensemble de chercheurs réputés, excellant dans leur domaine de recherche, desquels j'ai pu beaucoup apprendre. Ce contexte m'a été particulièrement bénéfique dans le sens ou au-delà de l'expérience acquise dans les disciplines du génie informatique, j'ai pu acquérir de bonnes connaissances en génie électrique. Ceci m'est essentiel puisque je souhaite que ma formation d'ingénieur soit avant tout généraliste.

Un autre point important et enrichissant de ce stage à été l'expérience du travail en autonomie. Les deux missions que j'ai effectué pour le projet CyDER ont été réalisées en autonomie ou parfois en binôme avec une grande liberté de prise de décision et d'initiative. Ce travail autonome et libre est une spécificité des environnements de recherche. Bien que l'expérience fut enrichissante, un désavantage notable a été la quasi-absence de management comme on en trouverait dans une entreprise industrielle. Faire l'expérience d'un management d'équipe de projet est une chose qui m'a manqué ici et que je souhaite trouver dans mon prochain stage.

Je ressors en fin de compte parfaitement satisfait de ce stage. Celui-ci a répondu aux attentes que j'avais d'une expérience de travail dans un laboratoire de recherche comme le LBNL. Ce stage a été une excellente valeur ajoutée à ma formation d'ingénieur.

7 Glossaire

- **LBNL**: Lawrence Berkeley National Laboratory
- **CyDER**: Cyber Physical Co-simulation Platform for Distributed Energy Resources in Smart Grids
- **FMI**: Functional Mock-up Interface
- **FMU**: Functional Mock-up Unit
- **API**: Application Programming Interface
- **Django**: Framework de développement web en Python, www.djangoproject.com
- **VueJS**: vuejs.org
- **PostgreSQL**: Base de données SQL utilisée par Django pour stocker les informations nécessaires au fonctionnement de l'interface web.
- **OpenStreetMap**: Base de données géographique libre, www.openstreetmap.org
- **Celery**: Bibliothèque Python qui permet de gérer des tâches de façon asynchrone en les organisant dans des files.
- **Gunicorn**: Une WSGI (Web Server Gateway Interface) qui fait le lien entre le projet Django et le serveur Nginx.
- **Leaflet**: Bibliothèque JavaScript open-source permettant de créer des cartes interactives sur une page web.
- **Nginx**: Un serveur HTTP.
- **Redis**: Base de données NoSQL utilisée comme messager pour Celery dans le projet d'interface web.
- **NSRDB**: National Solar Radiation Data Base, maps.nrel.gov/nsrdb-viewer/
- **NREL**: National Renewable Energy Laboratory, www.nrel.gov
- **Jupyter Notebook**: jupyter.org

- **Pandas**: pandas.pydata.org
- **Numpy**:www.numpy.org
- **Matplotlib**: matplotlib.org
- **Seaborn**: seaborn.pydata.org
- **SciKit-Learn**: Bibliothèque libre Python, très performante et particulièrement bien documentée, dédiée au Machine Learning, scikit-learn.org