Chapitre4 : Corrélations de mesures électriques

Wilfried Ehounou

July 6ᶜ 2018

Contents

Chapter 1

Mesures : Des Series Temporelles

Les mesures sur les arcs forment des séries temporelles. Certains arcs partagent des équipements que nous souhaitons déterminer à partir des séries temporelles. Nous supposons que les séries temporelles associées à ces arcs ont les mêmes comportements au cours du temps. En d'autres termes, toute variation dans une série est visible dans une autre série. Nous disons que ces arcs sont *corrélés* et la valeur liée à cette relation entre les arcs est désignée par *coefficient de similarité*. Dans le chapitre précédent, nous avons modélisé les mesures sur les arcs par des séries temporelles et la particularité de ces séries est la présence de valeurs érronnées et manquantes.
Le problème est de savoir s'il existe une méthode de calcul du coefficient de similarité qui tienne compte des erreurs dans les séries temporelles et qui vérifie l'hypothèse sus-jacente.
Pour ce faire, nous procédons comme suit : la première partie énonce les analyses sur les séries temporelles. La seconde partie présente les différentes méthodes de calcul des coefficients de similarité entre les séries. Et enfin la dernière partie sélectionne la méthode de calcul du *coefficient de similarité* puis analyse les performances de cette méthode sur les données réelles d'un *sous-réseau du datacenter Champlan.*

1.1 Séries temporelles

Définition 1 *Une série temporelle est une suite chronologique de valeurs réelles x_t à des instants de temps régulièrement espacés.*

$$(x_t)_{t\in\Theta} \tag{1.1}$$

avec Θ l'ensemble discret des espaces de temps de dimension n.

L'intervalle de temps entre deux mesures successives dépend de la série. Il peut s'agir d'un jour, d'une semaine, d'une minute, etc. Généralement, les séries temporelles sont utilisées pour comprendre les mécanismes qui produisent ces observations. Ces mécanismes sont associés au temps et permettent de faire les analyses suivantes :

- La modélisation : la représentation de la série sous la forme d'une fonction du temps.
- La prévision : prédire les données futures à partir de valeurs prédécentes.
- La détection de rupture : la série change-t-elle significativement à un instant t.
- La comparaison : déterminer la relation existante entre une série observée et d'autres séries candidates.

Nous décrivons brièvement les modèles d'analyses sur des séries temporelles puis nous présentons l'objectif recherché par l'analyse de ces séries dans notre étude.

1.1.1 La modélisation et la prévison d'une série temporelle

Un modèle est une image simplifiée de la réalité qui vise à traduire le fonctionnement d'un phénomène et permet de mieux les comprendre. Nous distinguons deux types de modèles :

- Les modèles déterministes : ils utilisent les éléments de la statistique descriptive et suppose que l'observation de la série à la date t est une fonction du temps t et d'une variable ϵ_t:

  $$x_t = f(t, \epsilon_t).$$

  La variable ϵ_t est le résidu ou l'erreur du modèle et représente la différence entre la réalité et le modèle proposé.
  Les deux modèles les plus utilisés sont les suivants :

  - Le modèle additif : c'est la décomposition de la série en trois termes

    $$x_t = Z_t + S_t + Q_t$$

    où Z_t est la tendance, S_t la périodicité et Q_t les composantes (erreurs) identiquement distribuées. Z_t, S_t sont aussi déterministes.
  - Le modèle multiplicatif : la variable x_t est le produit de la tendance et d'une composante périodique :

    $$x_t = Z_t(1 + S_t)(1 + Q_t).$$

    Toutefois, l'application d'un logarithme nous permet de revenir au modèle additif.

    $$x_t = log(Z_t) + log(1 + S_t) + log(1 + Q_t).$$

- Les modèles stochastiques : ils font l'hypothèse que les résidus ϵ_t ne sont pas indépendants et qu'il est possible de prévoir les résidus en partie. L'avantage réside dans la réduction de l'imprécision de la prévision des termes futures de la série temporelle. La variable ϵ_t devient une fonction des valeurs du passée et d'un terme d'erreur η_t

  $$\epsilon_t = (\epsilon_{t-1}, \epsilon_{t-2}, ..., \eta_t).$$

  Nous pouvons citer, comme exemple de modèles couramment utilisés, les modèles SARIMA, ARIMA et ARMA. Dans ces modèles, la modélisation porte sur la forme du processus (ϵ_t). Un exemple de modélisation est le modèles autorégressifs linéaires d'ordre 2 avec des coefficients autorégressifs a_1, a_2 défini par

  $$\epsilon_t = a_1 x_{t-1} + a_2 x_{t-2} + \eta_t,$$

  où η_t est un bruit blanc.

Un exemple de série temporelle et de ses différents composantes est présenté dans la figure 1.1. La série temporelle a la donnée du trafic des routes françaises de 1989 à 1996. La première ligne est la répresentation de la série temporelle et la seconde est celle de la tendance. Quant à la troisième

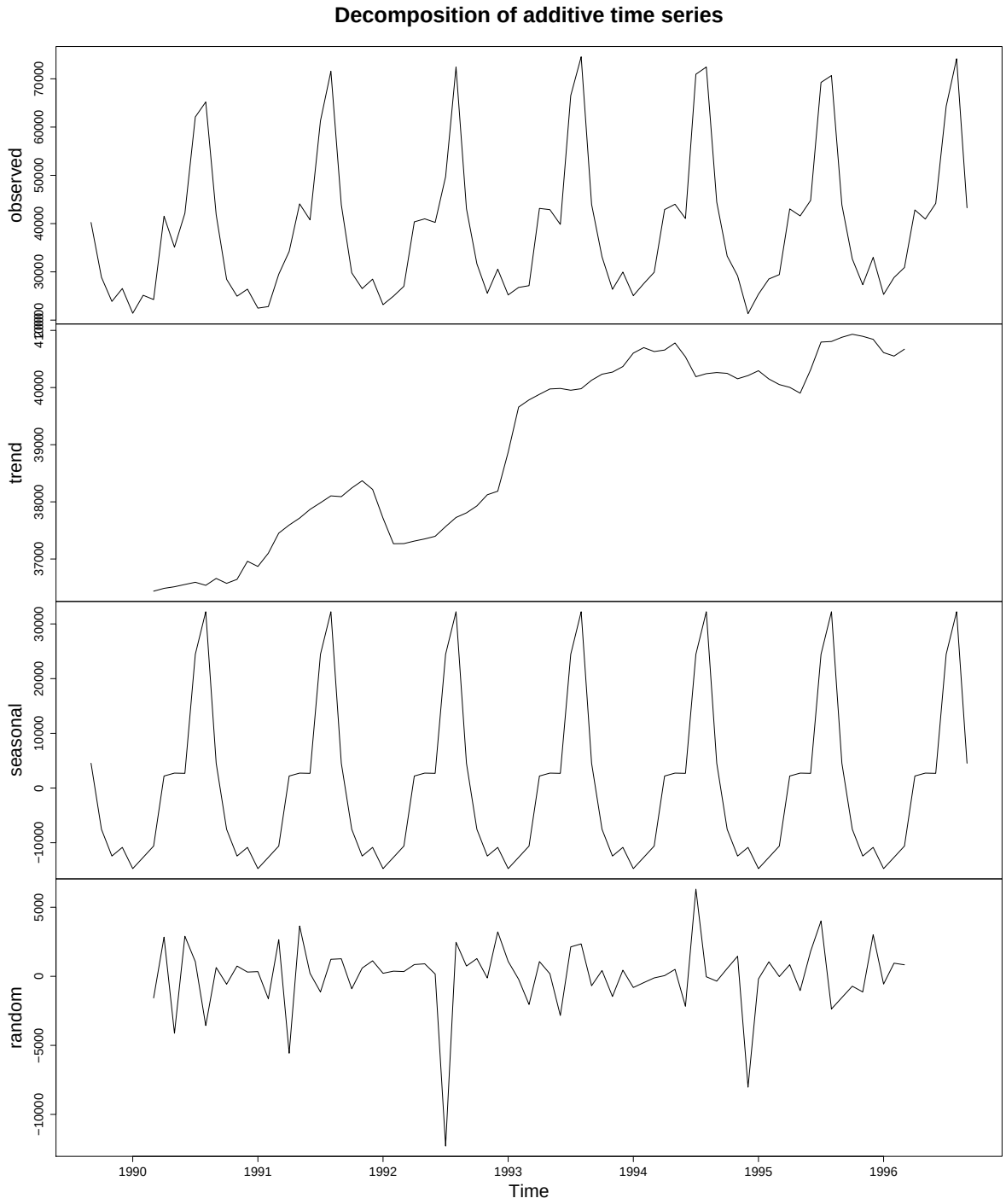

Figure 1.1: série temporelle du trafic routier français 1989 à 1996 avec ses composantes.

et la dernière ligne, elles représentent respectivement la figure de périodicité et les résidus. La période de la série est de 12.

Les deux types de modèles ci-dessus induisent des techniques de prévision bien particulières. Schématiquement, ils isolent la tendance et à la saisonnalité éventuelle. Puis ils cherchent à les modéliser et à les estimer. Enfin ils les éliminent de la série : ces deux opérations sont nommées la *détendancialisation* et la *désaisonnalisation* de la série. Une fois ces composantes éliminées, on

obtient la série aléatoire ϵ_t :

- Pour les modèles déterministes, cette série est considérée comme décorrélée et il n'y a plus rien à faire.
- Pour les modèles stochastiques, on obtient une série stationnaire (ce qui signifie que les observations successives de la série sont identiquement distribuées mais pas nécessairement indépendantes) qu'il s'agit de modéliser.

1.1.2 La détection de rupture

La détection de ruptures consiste à déceler la présence d'un ou de plusieurs pics dans la série temporelle et à les localiser dans la série temporelle. Déterminer l'existence d'une rupture est d'autant plus difficile que cette dernière n'est pas forcément caractérisée par un décalage de grande amplitude entre x_t et x_{t+1} par rapport à la dispersion des observations. Un enjeu de la détection est donc d'être sensible aux faibles variations tout en garantissant une certaine robustesse au bruit. La thèse [**?**] propose une méthode de détection de changements univariée et multivariée à la médiane des segments, le segment étant une subdivision de la série temporelle. Les modèles proposés sont exprimés par des fonctions de vraisemblance afin d'illustrer l'augmentation de la difficulté lors de l'ajout de nouvelles inconnues.

1.1.3 La comparaison de séries temporelles

Si deux séries sont observées, nous pouvons nous demander quelle influence elles exercent l'une sur l'autre. Par exemple, étant donnée deux séries X_t et Y_t, nous vérifions s'il existe par exemple des relations du type $Y_t = a_1 \times X_{t+1} + a_3 \times X_{t+3}$.
Ici, deux questions se posent : tout d'abord, la question de la causalité c'est-à-dire quelle variable (ici (X_t)) va expliquer l'autre (ici (Y_t)), ce qui conduit à la deuxième question, celle du décalage temporel : si une influence de (X_t) sur (Y_t) existe, avec quel délai et pendant combien de temps la variable explicative (X_t) influence-t-elle la variable expliquée (Y_t) ?

Nous allons considérer un réseau de flots modélisé par un graphe G et les arcs portent des mesures. Les mesures sont modélisées par des séries temporelles. Dans notre cas d'étude, nous cherchons à comparer des séries temporelles en supposant que les variations dans une série sont observables dans une autre série. Pour ce faire, nous définissons la corrélation entre des arcs comme suit :

Définition 2 *: corrélation entre arcs*
Soit corr le coefficient de similarité entre les séries x et y défini de $\mathbb{R}^n \times \mathbb{R}^n \to [0,1]$ et les arcs A et B contenant respectivement les séries x et y.
Deux arcs A et B sont corrélés si et seulement si le coefficient de similarité entre les séries x et y est contenu dans $[0.5, 1]$. ($corr(x,y) \in [0.5,1]$)

On dit alors que A et B sont *fortement corrélés* si le coefficient de similarité tend vers 1 ($corr(x,y) \to 1$) et *faiblement corrélés* si $corr(x,y)$ tend vers 0.5 ($corr(x,y) \to 0.5$).

Notre objectif est de trouver la méthode qui calcule au mieux la corrélation entre des arcs en tenant compte des caractéristiques de nos données (valeurs manquantes et érronnées à certains instants t dans les séries temporelles des arcs).

1.2 État de l'art des méthodes de similarité

Les différentes méthodes de calculs des coefficients de similarité se basent sur les séries temporelles associées aux arcs du réseau. Nous allons présenter les principales méthodes de calculs de similarité qui sont regroupées en 3 familles et qui sont détaillées dans cette section.

1.2.1 Similarité sur les séries entières

Les séries entières sont considérées comme des vecteurs et comparées avec une distance qui utilise toutes les valeurs des séries. Cette distance calcule la similarité entre ces deux séries. La similarité entre deux séries entières est excellente quand il existe des caractéristiques discriminatoires identiques entre ces séries sur l'axe du temps. Ces caractéristiques peuvent être identifiées aux mêmes instants de temps ou à des instants décalées mais constants dans le temps. Par exemple, considérons le problème *FiftyWord* [**?**] dans lequel les données proviennent de la base de donnée UCR [**?**]. Dans la figure 1.2, nous distinguons 4 séries regroupées en 2 classes. Les deux séries du haut (en noir) identifient la classe 30 et les deux séries du bas (en vert) identifient la classe 50. Le motif commun est observable en alignant les séries.

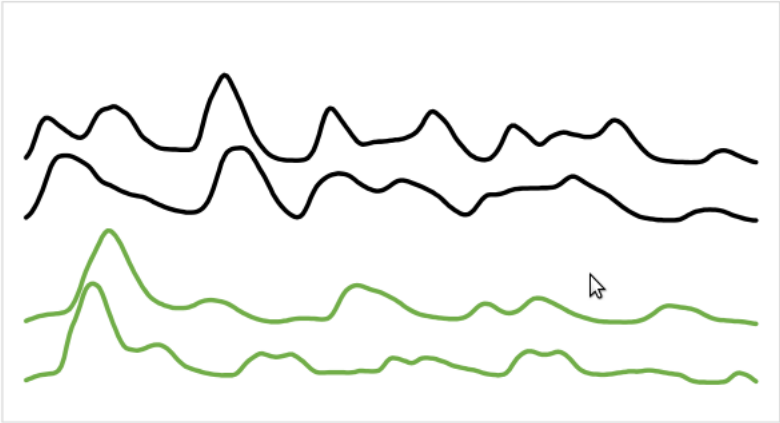

Figure 1.2: Détection du motif commun en alignant les séries. Les deux séries du haut représentent la classe 30 et les deux séries du bas représentent la classe 50.

Les approches basées sur les vecteurs sont pertinentes quand il existe une décalage de temps entre les pics et les creux des séries, comme c'est le cas avec les deux courbes du bas dans la figure 1.2.
Les méthodes telles que *Time Warp Edit*, *Move-Split-Merge*, *Longest Common Subsequence* et *distance de Pearson* sont des distances de *mesures élastiques* qui se servent de l'approche vectorielle. Les distances de *mesures élastiques* sont les meilleures approches pour traiter les problèmes des séries entières à savoir la détection de pics, de décalage et de creux entre les séries.

1.2.1.1 Dynamic Time Warping DTW

Dynamic time warping (DTW) est une technique pour trouver l'aligmenent optimal entre deux séquences dépendant du temps sous certaines contraintes (figure 1.3). Les séries temporelles sont déformées par une transformation non-linéaire de la variable temporelle, pour déterminer une mesure de leur similarité, indépendamment de certaines transformations non-linéaires du temps.

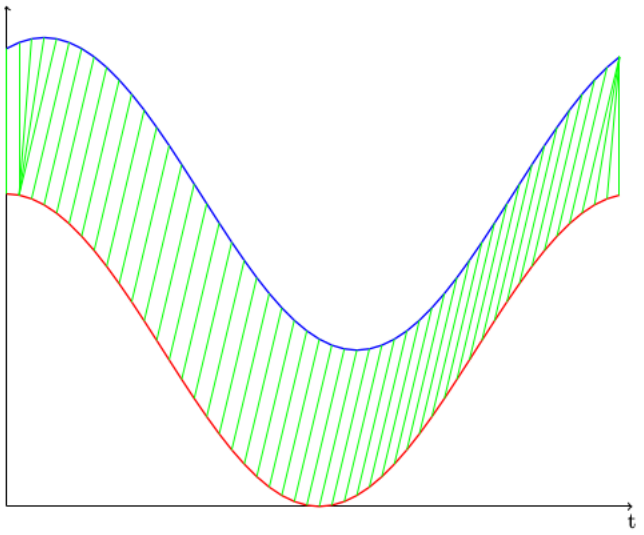

Figure 1.3: Deux séquences en dimension 1 alignées avec Dynamic Time Warping. Les coordonnées de la séquence du haut et de celle du bas correspondent respectivement à $cos(t)$ et à $cos(t + \alpha)$. Pour des questions de visualisation, la séquence du dessus a été décalée vers le haut lors du tracé. [**?**]

Supposons que nous souhaitons mesurer la similarité entre deux séries $A = (a_1, \cdots, a_m)$ et $B = (b_1, \cdots, b_m)$. Soit $M(A, B)$ la matrice de distances entre A et B où $M_{i,j} = (a_i - b_j)^2$. Un chemin d'alignement *Dynamic Time Warping (DTW)* est une méthode inspirée de la distance de *Levenshtein*, également appelé *distance d'édition*. À l'origine, *DTW* était appliqué dans le domaine de la reconnaissance vocale et permet de trouver l'alignement global optimal entre deux séquences, c'est-à-dire faire correspondre chaque élément de chaque séquence à au moins un élément de l'autre séquence en minimisant les coûts d'association. Le coût d'une association correspond à la distance entre les deux éléments; classiquement une l_p norm [**?**]. La figure 1.3 représente un exemple d'alignement opéré par *DTW*. Il illuste l'alignement de deux sinusoides légèrement déphasées. Le résultat numérique fournit par *DTW* correspond à la somme des hauteurs des "barreaux" formés par les associations. Les extrémités des alignements de la figure 1.3 montrent que *DTW* est capable de réaligner correctement une séquence par rapport à une autre, et parvient ainsi à saisir des similarités que la distance euclidienne ne peut extraire.
La distance *Dynamic Time Warping* est définie récursivement par :

$$D(A_i, B_j) = \delta(A_i, B_j) + min \begin{cases} D(A_{i-1}, B_{j-1}), \\ D(A_i, B_{j-1}), \\ D(A_{i-1}, B_j)) \end{cases}$$

où A_i représente la sous-séquence $(a_1, \cdots, a_i)$. Le coût de l'alignement optimal est alors donné par $D(A_{|A|}, B_{|B|})$. Le principe de programmation dynamique peut alors se résoudre par un arbre en partant des feuilles en supposant que le problème principal peut être symbolisé par la racine et les sous-problèmes par des noeuds appartenant aux différents sous-arbres. La fonction DTW peut alors être mémoisée : les différents appels peuvent être retenus afin de ne pas calculer deux fois la fonction appelée avec les mêmes paramètres. Aussi est-il habituel, comme l'arbre contient $|A|.|B|$ noeuds différents, de stocker ces différents résultats intermédiaires dans une matrice $|A| \times |B|$. Le calcul de *DTW* consiste alors à trouver le chemin de coût minimum dans la matrice, ce qui

s'exécute avec une complexité en temps et en mémoire de $\Theta(|A| \times |B|)$.

1.2.1.2 Time Warp Edit TWE

La distance *Time Warp Edit* (TWE) est une mesure de distance pour des séries temporelles discrètes. En comparaison avec les distances telles *Dynamic Time Warping (DTW)* [**?**] et *longest common subsequence (LCS)* [**?**], la distance *TWE* est une métrique proposée par *P.F. Marteau* en 2009. La distance *TWE* a quatre propriétés :

- Traite le décalage temporelle local avec des performances élevées.
- Satisfait l'inégalité triangulaire c'est-à-dire $AB \leq AC + CB$ avec AB, AC, CB des cotés d'un triangle quelconque.
- Comprend le paramètre de rigidité ν qui contrôle l'élasticité de la métrique.
- Utilise la différence de temps pour comparer les segments de séries temporelles comme des coûts de correspondances locales.

Le paramètre ν est important dans l'algorithme de *TWE* parce qu'il rend plus flexible l'identification de motifs (matching) entre les séries temporelles. La preuve de son efficacité a été prouvée dans [**?**].

L'algorithme de *TWE* introduit trois opérations : $delete_A$, $delete_B$ et *match* pour l'édition de deux séries discrètes A et B. L'édition d'une série temporelle consiste à modifier cette série en sous-ensembles de mêmes tailles à partir d'une des trois opérations et chaque sous-ensemble est désigné par *séquence*. La similarité entre les séries A et B est le coût minimum de séquences nécessaire pour transformer A en B. En se basant sur ces trois opérations, l'algorithme de *TWE* calcule le coût d'une séquence à chaque opération pour toutes les paires de séries avec l'équation 1.2. Dans cette équation, U désigne l'ensemble des séries finies $U = \{A_1^p \mid p \in \mathbb{N}\}$ où A_1^p est une série avec des temps discrets variant de 1 à p, A_1^0 est la série nulle de longueur nulle et a'_i désigne le i^{ieme} élement de la série A. Nous considérons alors que $a'_i \in S \times T$ où $S \subset R^d$ avec $d \geq 1$ intègre un espace multidimensionnel de variables et $T \subset R$ intègre la variable temporelle. Ainsi, nous pouvons écrire $a'_i = (a_i, t_{a_i})$, où $a_i \in S$ et $t_{a_i} \in T$ avec la condition que $t_{a_i} > t_{a_j}$ quand $i > j$ (le temps est strictement croissant dans la séquence des éléments). La pénalité notée λ est constante. La similarité entre les séries A et B est calculée recursivement par

$$\delta_{\lambda,\nu}(A_1^p, B_1^q) = min \begin{cases} \delta_{\lambda,\nu}(A_1^{p-1}, B_1^q) + \Gamma(a'_p \to \Lambda) & delete_A, \\ \delta_{\lambda,\nu}(A_1^{p-1}, B_1^{q-1}) + \Gamma(a'_p \to b'_q) & match, \\ \delta_{\lambda,\nu}(A_1^{p-1}, B_1^q) + \Gamma(\Lambda \to a'_p) & delete_B, \end{cases} \tag{1.2}$$

avec

$$\begin{array}{lcl} \Gamma(a'_p \to \Lambda) & = & d(a'_p, a_{p-1}) + \lambda \\ \Gamma(a'_p \to b'_q) & = & d(a'_p, b'_q) + d(a'_{p-1}, b'_{q-1}) \\ \Gamma(\lambda \to b'_q) & = & d(b'_q, b'_{q-1}) + \lambda. \end{array}$$

La récursivité est initialisée par

$$\begin{cases} \delta_{\lambda,\nu}(A_1^0, B_1^0) = 0, \\ \delta_{\lambda,\nu}(A_1^0, B_1^j) = \infty \ pour \ j \geq 1, \\ \delta_{\lambda,\nu}(A_1^0, B_0^j) = \infty \ pour \ j \geq 1, \end{cases}$$

avec $a'_0 = b'_0 = 0$ par convention.
L'algorithme *TWE* introduit la rigidité, constante positive ν, dans la définition de $\delta_{\lambda,\nu}$ en choisissant $d(a', b') = d_{LP}(a, b) + \nu \times d_{LP}(t_a, t_b)$ qui caractérise la rigidité de $\delta_{\lambda,\nu}$. Si $\nu = 0$ alors $\delta_{\lambda,\nu}$ est la distance de S et non de $S \times T$. Dans cette équation, d_{LP} est la métrique l_p norm [**?**]. L'expression finale de $\delta_{\lambda,\nu}$ est présentée par l'équation 1.3.

$$\delta_{\lambda,\nu}(A_1^p, B_1^q) = min \begin{cases} \delta_{\lambda,\nu}(A_1^{p-1}, B_1^q) + \Gamma(a'_p \rightarrow \Lambda) & delete_A, \\ \delta_{\lambda,\nu}(A_1^{p-1}, B_1^{q-1}) + \Gamma(a'_p \rightarrow b'_q) & match, \\ \delta_{\lambda,\nu}(A_1^{p-1}, B_1^q) + \Gamma(\Lambda \rightarrow a'_p) & delete_B, \end{cases} \tag{1.3}$$

avec

$$\begin{array}{lcl} \Gamma(a'_p \rightarrow \Lambda) & = & d_{LP}(a'_p, a_{p-1}) + \nu.(t_{a_p} - t_{a_{p-1}}) + \lambda \\ \Gamma(a'_p \rightarrow b'_q) & = & d_{LP}(a'_p, b'_q) + d_{LP}(a'_{p-1}, b'_{q-1}) + \nu.(|t_{a_p} - t_{b_q}| + |t_{a_{p-1}} - t_{b_{q-1}}|) \\ \Gamma(\lambda \rightarrow b'_q) & = & d_{LP}(b'_q, b'_{q-1}) + \nu.(t_{b_q} - t_{b_{q-1}}) + \lambda \end{array}$$

1.2.1.3 Move-Split-Merge MSM

Move Split Merge (MSM) est une distance qui est basée sur le coût de transformation d'une série temporelle en une autre série en utilisant une séquence d'opérations de *move*, *split* et *merge*. *MSM* a l'avantage d'être une métrique et être invariant au choix de l'origine de la série. En effet, soit $X = (x_1, \cdots, x_m)$ une série temporelle dans laquelle x_i est un réel. Une translation de X par t, où t est aussi un réel, est une transformation qui ajoute t à chaque élement de X et produit la série $X + t = (x_1 + t, \cdots, x_m + t)$. Si la distance D est invariant au choix de l'origine alors $D(X, Y) = D(X + t, Y + t)$. *MSM* est invariant au choix de l'origine parce que toute transformation S qui convertit X et Y convertit aussi $X + t$ en $Y + t$. Quant à la métrique, elle permet l'utilisation d'un large nombre d'outils pour indexer, regrouper et visualiser des séries dans des espaces vectoriels arbitraires.
Nous présentons un algorithme de complexité quadratique qui calcule la similarité entre deux séries. Soient $X = (x_1, \cdots, x_m)$ et $Y = (y_1, \cdots, y_n)$ deux séries temporelles. L'algorithme 1 décrit la méthode dynamique de calcul de la distance entre X et Y. Pour chaque couple (i, j) tel que $1 \leq i \leq m$ et $1 \leq j \leq n$, nous définissons $Cost(i, j)$ la distance *MSN* entre les i premiers éléments de X et les j premiers éléments de Y. Ainsi, la distance *MSN* entre X et Y est simplement $Cost(X, Y)$. Comme indiqué dans l'algorithme 1, $Cost(i, j)$ est calculé récursivement en se basant sur $Cost(i, j-1)$, $Cost(i-1, j)$ et $Cost(i-1, j-1)$.

$$Cost(i, j) = min \begin{cases} Cost(i-1, j-1) + |x_i - y_j|, \\ Cost(i-1, j) + C(x_i, x_{i-1}, y_j), \quad \text{avec} \\ Cost(i, j-1) + C(y_j, x_i, y_{j-1}) \end{cases}$$

$$C(x_i, x_{i-1}, y_j) = \begin{cases} c \ si \ x_{i-1} \leq x_i \leq y_j \ ou \ x_{i-1} \geq x_i \geq y_j \\ c + min(|x_i - x_{i-1}|, |x_i - y_j|) \ sinon \end{cases}$$

1.2.1.4 Distance de Pearson

Dans la comparaison de séries temporelles, nous avons toujours constaté que la normalisation dans le calcul d'une distance euclidienne entre des séries donne de meilleurs résultats. Nous allons montré la relation existance entre la distance euclienne et la coefficient de Pearson.

Algorithm 1 MSM(X,Y)

Require: série $X = (x_1, \cdots, x_m)$
Require: série $Y = (y_1, \cdots, y_n)$
$Cost(1,1) = |x_1 - y_1|$
for $i = 2, \cdots, m$ **do**
    $Cost(i,1) = Cost(i-1,1) + C(x_i, x_{i-1}, y_1)$
end for
for $j = 2, \cdots, n$ **do**
    $Cost(1,j) = Cost(1,j-1) + C(y_j, x_1, y_{j-1})$
end for
for $i = 2, \cdots, m$ **do**
    for $j = 2, \cdots, n$ **do**

$$Cost(i,j) = min \begin{cases} Cost(i-1,j-1) + |a_i - b_j|, \\ Cost(i-1,j) + C(a_i, a_{i-1}, b_j), \\ Cost(i,j-1) + C(b_j, a_i, b_{j-1}) \end{cases}$$

    end for
end for
return la distance MSM $D(X,Y)$ est $Cost(m,n)$.

Soient deux séries temporelles A et B composées de T éléments $A = (a_1, \cdots, a_T)$ *et* $B = (b_1, \cdots, B_T)$. La distance euclidienne entre A et B est définie par l'équation 1.4

$$d_E = \sum_{t=1}^{T} (a_i - b_i)^2 \tag{1.4}$$

Elle est une métrique et les séries A et B sont identiques si cette distance est égale à 0. Pour les analyses de séries, il est recommendé de normaliser les séries pour éviter les variations d'échelles. En ce qui concerne le coefficient de Pearson, il mesure la corrélation ϱ entre deux variables aléatoires X et Y comme indiqué dans l'équation 1.5.

$$\varrho_{X,Y} = \frac{E[X - \mu_X][Y - \mu_Y]}{\sigma_X . \sigma_Y} \tag{1.5}$$

où μ_X est la moyenne et σ_X est l'écart-type de X. Le coefficient $|\varrho| = 1$ est égal à 1 si X et Y sont parfaitement corrélées et 0 si X et Y sont non corrélées.
Dans le but d'utiliser le coefficient de Pearson comme une distance de séries temporelles, nous introduisons la *distance de Pearson* en générant de petites valeurs de distances pour des séries similaires. Elle est définie par l'équation 1.6.

$$d_P(A,B) = 1 - \varrho_{A,B} = 1 - \frac{\frac{1}{T}\sum_{t=1}^{T}(a_t - \mu_A)(b_t - \mu_B)}{\sigma_A . \sigma_B} \tag{1.6}$$

avec $0 \leq d_P(A,B) \leq 2$. Nous obtenons une parfaite correspondance ($d_P = 0$) pour les séries A et B s'il existe des nombres $\alpha, \beta \in \mathbb{R}$ avec $\beta > 0$ tel que $a_i = \alpha + \beta * b_i$. Nous exprimons d_E en fonction de d_P.

$$d_E(A,B) = \sum_{t=1}^{T}(a_i - b_i)^2 = \sum_{t=1}^{T}(a_i - 0)^2 - 2\sum_{t=1}^{T}(a_i . b_i) + \sum_{t=1}^{T}(b_i - 0)^2 \tag{1.7}$$

Les termes $\sum_{t=1}^{T}(a_i - 0)^2$ et $\sum_{t=1}^{T}(b_i - 0)^2$ correspondent à l'écart-type des séries A et B en supposant que la moyenne de ces séries est nulle $\mu_A = \mu_B = 0$ et leurs écarts-types sont égaux à 1 ($\sigma_A = \sigma_B = 1$). L'équation précédente 1.7 devient

$$d_E(A_{norm}, B_{norm}) = 2.T(1 - \frac{\frac{1}{T}\sum_{t=1}^{T}(a_{i,norm} - 0)(b_{i,norm} - 0)}{1.1}) = 2.T.d_P(A_{norm}, B_{norm})$$

Ainsi, la distance euclidienne de deux séries normées est égale à la distance de Pearson multiplié par un facteur constant $2T$. L'équivalence entre ces deux distances est pertinente parce que certains algorithmes utilisent la distance euclidienne pour faire de la classification (cas de k-means). Dans l'article [**?**], l'auteur utilise la classification *k-means* de séries avec la distance de Pearson et il en conclut que cette classification donne les mêmes résultats que le *k-means* avec la distance euclidienne.

1.2.1.5 Longest Common Subsequence

La distance *longest common subsequence (LCSS)* est basée sur la reconnaissance de motifs dans les problèmes *(LCSS)* . Dans ces problèmes, il recherche la plus longue séquence qui est commune aux deux séries discrètes en utilisant la distance *edit distance (ED)*. Cette approche a été élargie aux séries continues en définissant la variable de seuil ϵ qui indique la différence entre une paire de valeurs. Cette différence détermine s'il existe une similarité entre ces séries. *LCSS* trouve un alignement optimal entre deux séries en insérant des écarts pour déterminer le nombre maximum de paires correspondantes. La distance *LCSS* entre deux séries A et B peut être calculée à partir de l'algorithme 2.

Algorithm 2 LCSS(A,B)

```
Soit L une matrice initialisée à 0 de dimension (m+1) × (m+1)
for i ← m a 1 do
    for j ← m a 1 do
        L_{i,j} = L_{i+1,j+1}
        if a_i = b_j then
            L_{i,j} ← L_{i,j} + 1
        else if L_{i,j+1} > L_{i,j} then
            L_{i,j} ← L_{i,j+1}
        else if L_{i+1,j} > L_{i,j} then
            L_{i,j} ← L_{i+1,j}
        end if
    end for
end for
return L_{1,1}
```

La distance *LCSS* entre les séries A et B est

$$d_{LCSS}(A, B) = 1 - \frac{LCSS(A, B)}{m}$$

Conclusion : plusieurs distances ont été presentées et leur point commun est l'utilisation de toutes la série pour le calcul de la similarité. Parmi ces distances, nous distinguons les distances qui sont des métriques *TWE, MSM et la distance de Pearson* et celles qui ne le sont pas *LCSS et DTW*.

1.2.2 Similarité sur les séquences

Nous présentons, dans cette section, les distances qui transforment au préalable les séries pour comparer leurs caractéristiques. Ces caractéristiques sont appelées des *features*.
Pour mesurer la similarité entre des séries temporelles, des méthodes proposent de répresenter les données en sous-séquences différentes, chacune formant une classe. Cette transformation est désignée par *shapelets*. Un shapelet considère que les sous-séquences sont indépendantes. Le calcul de similarité entre des séries se produit localement entre les séquences de même phase au moyen d'une métrique. Généralement on utilise la distance euclidienne. D'abord, les shapelets généralisent l'algorithme des k plus proche voisins (*k-means*) largement utilisé dans les arbres de décision pour améliorer les classifications [**?**]. Ensuite, ils sont interprétables et donne une idée de la différence entre deux classes [**?**]. Enfin, ils peuvent être très précis que d'autres méthodes concurrentes [**?**, **?**].
Hills et al. [**?**] propose l'algorithme 3 de transformation de séries en *shapelets* en retournant les k premiers shapelets dans une seule exécution. L'algorithme 3 se décrit comme suit: Soient w_i une sous-série d'une série temporelle A avec $i \leq k$ et W_l l'ensemble de taille l de séries w_i. La distance de *shapelet* $sDist(S,T)$ entre un shapelet S et une série T est la distance euclidienne minimum entre S et $w_i \in W_l$.

$$sDist(S,T) = min_{w_i \in W_l}(dist(S, w_i))$$

Le meilleur shapelet a une distance $sDist$ faible pour les instances d'une classe et des distances $sDist$ élevées pour les instances des autres classes.
Nous considérons w_i comme un *shapelet candidat*. L'ensemble de valeurs de $sDist$ pour chaque candidat est trouvé en utilisant la fonction *findDistances* et est évalué par la procédure *assessCandidate* au moyen de la mesure $f - statistic$. Les k meilleurs shapelets sont retournés après la suppression des sous-séries candidats par la fonction *removeSelfSimilar*. Nous nous servons de la procédure d'estimation de longueur, décrite dans [**?**], pour trouver les valeurs appropriées à utiliser comme les longueurs maximales et minimales de shapelets. Nous générons un maximum de $k = 10n$ shapelets où n est la taille de la série initiale.
Nous transformons la série initiale en utilisant les meilleurs shapelets comme des features où $sDist(S_i, T_j)$ désigne l'élément i dans l'instance j de la série transformée, S_i est le i^{ieme} shapelet et T_j est la j^{ieme} instance dans la série initiale. La complexité de l'algorithme est de $\mathcal{O}(n * m^2)$ avec n le nombre de séries temporelles et m la plus longue série temporelle [**?**].

Conclusion : les shapelets transforment la série en un sous-ensembles de séquences avec la fonction *generateCandidate*. Chaque séquence d'une série est nommée *shapelet candidat* et ce *shapelet candidat* est comparé avec les autres shapelets candidat de la même série à partir à la *distance de shapelet*. Une fois les *shapelets candidats* de chaque série selectionnée avec l'algorithme *ShapeletSelection*, on les compare avec la distance euclidienne. Cette méthode est inéfficace pour de séries de grandes tailles.

1.2.3 Similarité par aggrégation des caractéristiques descriptives

Les modèles de classification de séries temporelles basés sur des caractéristiques descriptives supposent d'extraire un ensemble de caractériques qu'on espère être représentatif de la forme générale d'une série temporelle. Le plus communément, ces caractériques sont quantifiées pour former des "sacs de mots" (BoW pour "Bag of Words"). Dans la récupération d'information, l'approche BOW d'estimation de la fréquence des mots en ignorant leur localisation est très commune. L'idée est d'estimer la fréquence d'occurences des caractéristiques des séries puis d'utiliser ces fréquences

Algorithm 3 ShapeletSelection(T,min,max,k)

```
1: kShapelets ← ∅
2: for all T_i in T do
3:     shapelets ← ∅
4:     for l ← min to max do
5:         W_{i,l} ← generateCandidates(T_i, l)
6:         for all subsequence S ∈ W_{i,l} do
7:             D_S ← findDistances(S, T)
8:             quality ← assessCandidate(S, D_S)
9:             shapelets.add(S, quality)
10:        end for
11:    end for
12:    sortByQuality(shapelets)
13:    removeSelfSimilar(shapelets)
14:    kShapelets ← merge(k, kShapelets, shapelets)
15: end for
16: return kShapelets.
```

comme des features pour faire de la classification.
Les approches suivantes diffèrent uniquement par les caractéristiques extraites. En effet, l'approche *Bag Of Pattern (BOP)* [**?**] convertit la série temporelle en une série discrète grâce à la méthode *Symbolic Aggregate approXimation (SAX)* [**?**]. Il crée un ensemble de mots *SAX* pour chaque série par l'application d'une fenêtre glissante, puis se sert de la fréquence des mots dans la série comme sa nouvelle caractéristique. *Baydoyan et al.* [**?**] décrit l'approche *bag-of-features* qui combine les caractéristiques de fréquences et d'intervalles. L'algorithme appelé *time series based on bag-of-features representation (TSBF)* implique la séparation entre la création de features et les étapes de classification. La création de features implique la génération d'intervalles aléatoires et les features représentent, généralement, la moyenne, la variance et la pente sur un intervalle. Le début et la fin d'un intervalle sont incluses dans les features.

1.2.4 Conclusion sur les méthodes de similarité

Dans cette partie, nous avons énuméré les différentes méthodes (distances) que nous pouvons utiliser pour calculer la similarité entre des séries. Nous avons catégorisé les distances en deux groupes : celles qui n'apportent aucune modification aux séries et celles qui transforment les séries avant de débuter l'analyse. La transformation des séries se fait de deux manières. La première consiste à diviser la série en séquences et à supposer que chaque séquence est indépendante des autres. Quant à la seconde, elle consiste à remplacer la série par certaines caractéristiques descriptives tel que la moyenne, l'écart-type, l'encodage de la série par des mots. Parmi celles qui ne modifient pas les séries, nous distinguons certaines qui ont la propriété de métriques. Cette propriété est importante pour choisir la méthode de calcul de la similarité entre des séries.

1.3 Méthode de similarité entre des mesures électriques : distance de Pearson

1.3.1 Choix de la distance

Nos séries temporelles ont la particularité d'être des mesures électriques. Ces séries sont associées aux arcs entrants dans des équipements. Pour certaines grandeurs comme la puissance ou l'intensité, les mesures se propagent dans l'ensemble du réseau en suivant la loi de conservation [**?**]. Cela implique que tout variation de consommation électrique des équipements apparait dans les courbes des séries temporelles comme des pics. Cela signifie que la consommation en puissance d'un équipement baisse quand les équipements qu'il alimente sont à l'arrêt ou augmente quand ces équipements se mettent en marche. Nous désignons la courbe d'une série temporelle par le *profil de consommation* d'un arc. Ainsi nous supposons que les arcs rattachés aux mêmes équipements ont les mêmes profils de consommation. Néammoins, des arcs appartenant à la chaine de propagation de l'électricité (c'est-à-dire tous les arcs par lesquels l'électricité transite) n'ont pas les mêmes profils car certains équipements sont rattachés à deux sources d'énergies et les pics s'attenuent durant la propagation. *La distance de similarité entre des paires de séries calcule le coefficient de similarité qui compare les profils de consommation des arcs.* Des arcs ont les mêmes profils si et seulement si le coefficient de similarité est le plus élevé (c'est-à-dire 1) et ont des profils différents si leur coefficient est le plus faible (c'est-à-dire 0).

Le choix de notre méthode de calcul de similarité dépend des données que nous avons collectées. En effet, dans ces données, nous avons certains arcs qui n'ont pas de mesures associées à des grandeurs physiques. Chaque valeur dans une série est la moyenne des valeurs sur un intervalle de 10 minutes. Il existe aussi des valeurs manquantes dans certaines séries de mesures. Nous avons attribué à ces valeurs manquantes la moyenne des valeurs à leur voisinage. Cette interpolation pose problème dans le calcul de similarité avec les *shapelets*. En effet, étant donnée le shapelet candidat contenant des valeurs extrapolées, la correspondance entre le shapelet candidat et toutes autres séquences est fortement dégradée à cause des valeurs extrapolées qui augmentent la distance euclidienne entre elles. Ensuite, il s'exécute lentement sur de grands ensembles de données. Enfin le shapelet candidat est de longueur quelconque et sa détermination passe par la génération de toutes les shapelets possibles. Ce qui conduit à une complexité de $\mathcal{O}(m^2)$, avec m la taille de la longue série temporelle. Cela rend le calcul impossible quand notre ensemble de données est de dimension $30 * 4320$ avec 30 le nombre de séries temporelles et 4320 le taille d'une série temporelle.

Par ailleurs, les méthodes par aggrégation des caractéristiques descriptives fournissent également des résultats mitigés. Prenons l'exemple de méthodes de similarité avec *Symbolic Aggregate approXimation*. Elle consiste à subdiviser chaque série en M séquences de tailles identiques puis à encoder chaque séquence par une lettre alphabétique, chaque lettre étant choisie dans un alphabet de lettres prédéfinies. La transformation d'une séquence en une lettre s'obtient grâce à une représentation *PAA (Piecewise Aggregate Approximation)* et à une table de correspondance entre l'alphabet. La représentation *PAA* [**?**] d'une séquence est la moyenne des valeurs de la séquence. La table de correspondance contient la liste ordonnée des points appelés *breakpoints* dont chaque valeur est une division arbritraire de la distribution gaussienne en zones équiprobables [**?**] et un alphabet. Et à chaque point de la liste *breakpoints*, une lettre lui est associée. La transformation de cette représentation en lettres a une complexité de $\mathcal{O}(mM)$ [**?**] avec M est le nombre de séquences de la série et m la taille de la série. Le principal inconvenient provient de l'erreur produite lors de

transformation. L'encodage considéré par *SAX* est celui qui minimise cette erreur. Ainsi, une série peut avoir deux encodages différents si nous changeons l'origine des séquences. L'encodage n'est donc pas unique. De même, il est difficile de détecter les variations dans la série avec l'encodage *SAX* car toute variation faible mais continue dans le temps a le même encodage. Par contre, une variation forte dans la série ne modifie pas l'encodage dans le *breakpoint* puisque la distance *PAA* est la moyenne des valeurs dans la séquence.

Enfin, les méthodes de similarité dont le résultat est le moins impacté par les valeurs extrapolées et les profils de consommation sont celles qui utilisent l'intégralité de séries temporelles. À cet effet, la fenêtre glissante de 6 (correspondant à une heure) permet d'abord d'attribuer des valeurs aux instants t ayant des valeurs manquantes puis de supprimer des petites variations qui peuvent pénaliser nos similarités et enfin de mettre en évidence les fortes variations dans la série. Parmi les distances énumérées dans la section 1.2.1, nous avons décidé de choisir la *distance de Pearson* comme méthode de calcul du coefficient de similarité entre les séries temporelles pour les raisons suivantes :

- La distance de Pearson est une métrique alors que *DTW* ne l'est pas. Toutefois, ces distances ne respectent pas l'inégalité triangulaire.
- La classification par la méthode de *k-means* avec la distance euclidienne produit les mêmes résultats que celle avec la distance de Pearson [**?**]. *k-means* est une classification de réference dans les bases de données *UCR* [**?**]
- Elle est de complexité *linéaire* alors que la complexité de la distance *MSM* est quadratique.
- Elle ne traite pas de décalage entre les séries comme *TWE*. Le traitement du décalage entre les séries n'est pas nécessaire puisque les séries sont moyennées sur 10 minutes et les possibles valeurs décalées ont déjà été moyennées. De même, nous ne sommes pas parvenus à trouver une valeur de rigidité ν correcte pour *TWE* à cause des problèmes dans le dataset de *Champlan* (valeurs manquantes et incorrectes).
- La distance de Pearson nécessite que les séries soient normalisées. Les coefficients de Pearson désignent les similarités entre des paires de séries normalisées et ils appartiennent à l'intervalle $[0, 1]$. Si le coefficient est égal à 1 alors il existe une forte similarité entre les séries. Cependant, il n'existe pas de similarité si le coefficient est égal à 0.

Conclusion : La distance de Pearson est la mieux adaptée aux calculs des coefficients de similarité entre des séries parce qu'elle est une métrique, de complexité linéaire, détecte les variations simultanées (faibles et fortes) entre des paires de séries. Toutefois, elle ne respecte pas l'inégalité traingulaire et nécessite que les données soient normalisées. Elle est donc une bonne métrique pour comparer les profils de consommation.

1.3.2 Résultats sur des données réelles

Nous présentons les résultats obtenus avec la distance de Pearson.
Les grandeurs physiques collectées proviennent du *datacenter Champlan* et sont au nombre de 10. Elles sont regroupées dans les systèmes triphasé et monophasé. Dans les séries de mesures provenant des arcs en triphasé, nous remarquons qu'il y'a toujours des mesures sur une phase et les autres phases ne contiennent aucune mesure. Il n'y a aucune mesure sur deux phases simultanément

et lorsqu'il existe des mesures, les i premières mesures sont sur la phase 1, les $n-i$ mesures sont sur la phase 2. Les autres grandeurs ne contiennent des mesures soit dans les i premiers instants de temps, soit dans les $n-i$ derniers instants de temps. Il est alors difficile d'exploiter ces mesures partielles. Quant au système monophasé, 20% des mesures des puissances réactives Q et apparentes S dans une série temporelle sont manquantes, 25% des mesures ne correspondent pas aux formules de calculs théoriques. Le nombre de valeurs à considérer dans les séries des grandeurs n'est pas assez significative pour calculer les similarités parce que la distance de Pearson est sensible à la dimension de la série à cause de sa relation avec le coefficient de Pearson. Il est difficile de trouver de possibles comportements identiques à partir des hypothèses de corrélation avec de séries de grandes tailles avec autant des valeurs absentes.
Nous considérons un sous-réseau électrique de *Champlan* dans lequel

- La grandeur sélectionnée possède des valeurs quelque soit le système (triphasé ou monophasé). La grandeur qui respecte cette règle est la *puissance* P.
- Les séries temporelles correspondent à un mois de mesures. Dans chaque série, nous avons en moyenne 15% de valeurs manquantes que nous remplacons par la valeur moyenne de chaque série.
- L'application de la loi des noeuds est possible.
- Aucun équipement n'est alimenté par un onduleur. La présence d'un onduleur modifie le profil de consommation d'un arc car l'onduleur recrée le signal en attribuant des puissances différentes. Cela implique que nous avons des mêmes variations dila baisse de puissance est compensée par l'onduleur.

Un exemple de ce sous-réseau est illustré dans la figure 1.4 dans lequel nous avons 31 équipements. Les sources sont identifiées par *TGBT* (TGBT1, TGBT2, TGBT4) et GF(1,2) désigne le groupe froid qui alimente la climatisation. Les tableaux sont *DD205, DD206, DD105, DD106, DD108, MSC3, R486, R481, CVC1 et CVC2*. Les baies sont *R491, R488, R484A, R484B, R042, R483, R487, R492, R490, R493, R494*. Les onduleurs sont indiqués par *OND* (1,2,RG). Les équipements *TGBT* sont alimentés par une source externe au datacenter, le fournisseur d'électricité régional *Enedis*.

Nous allons calculé la corrélation entre les arcs du *sous-réseau de Champlan* en supposant que *deux arcs incidents à un équipement sont fortement corrélés et deux arcs non corrélés ne possèdent aucun équipement en commun*. Chaque arc est identifié par deux équipements, celui qui fournit l'électricité (x) et celui qui en consomme (y). Nous désignons par convention $x->y$, l'arc entre x et y. Par exemple, l'arc entre $TGBT1$ et $R481$ est noté $TGBT1->R481$.
Soient trois arcs $x->y$ et $y->z$ et $t->u$. Le coefficient de similarité $corr(x->y, y->z) = 1$ signifie que les arcs $x->y$ et $y->z$ ont les mêmes profils de consommation, partagent un sommet c'est-à-dire y et sont fortement corrélés. De même, le coefficient $corr(x->y, t->u) = 0$ indique que les arcs $x->y$ et $t->u$ ne sont pas corrélés c'est-à-dire qu'ils n'ont pas de sommet en commun et que leurs profils de consommation sont différents.

Nous disposons de la topologie électrique réelle de *Champlan*. Nous comparons les coefficients de similarité calculés par rapport aux arcs qui sont incidents dans la topologie de *Champlan*. Pour ce faire, nous présentons, dans la figure 1.5, les distributions des arcs incidents et non incidentes en fonction des coefficients de similarité. Les arcs non incidentes sont désignés par *cases_0* et les

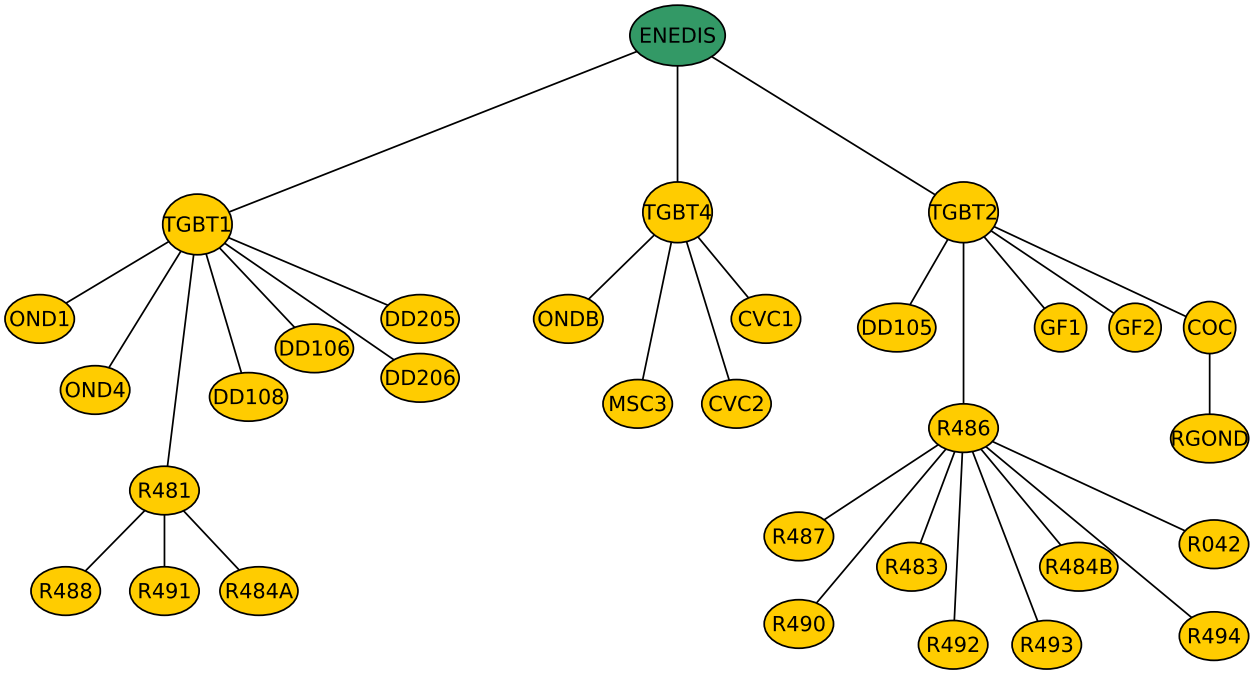

Figure 1.4: Les sources sont *TGBT1, TGBT2, TGBT4*. $GF(1,2)$ désigne le groupe froid qui alimente la climatisation. Les tableaux sont *DD205, DD206, DD105, DD106, DD108, MSC3, R486, R481, CVC1 et CVC2*. Les baies sont *R491, R488, R484A, R484B, R042, R483, R487, R492, R490, R493, R494*. Les onduleurs sont indiqués par *OND1,OND2,RGOND*. Les équipements *TGBT* sont alimentés par une source externe au datacenter, le fournisseur d'électricité régional *Enedis*.

arcs incidents par *cases_1*. La distribution des arcs non incidents est asymétrique vers la gauche. Cela est normale puisque nous avons supposé que les arcs non incidents ont des coefficients de similarité qui tendent vers 0. Cependant, nous avons 4 paires d'arcs qui ont des coefficients $corr(x->y, y->z) = 1$. La figure 1.6 présente les profils de consommation de ces 4 paires arcs. Nous constatons que :

- Les arcs $R486->R487$ et $R481->R488$ ont des profils opposés et en appliquant la valeur absolue sur les valeurs de chaque série, les profils se superposent. Il n'y a aucune corrélation entre cette paire d'arcs.
- Les arcs $TGBT1->DD205$ et $TGBT4->MSC3$ ont leurs profils qui se superposent et la corrélation de Pearson entre ces séries est alors nulle.
- Les paires d'arcs $TGBT2->GF2$ et $TGBT1->DD106$ ont des courbes qui ont plusieurs points d'intersection. À ces points, le coefficient de similarité est nul et est proche de 0 au voisinage de ces points. La corrélation de Pearson entre ces séries est alors nulle.

Cela implique que le coefficient de similarité est égal à $corr(x->y, y->z) = 1$ entre ces paires d'arcs $x->y, y->z$ par l'équation 1.6 alors que ces arcs n'ont aucun sommet en commun dans le graphe de la figure 1.4. Ces erreurs de corrélation sont introduites par la méthode de calcul des coefficients.

En ce qui concerne la distribution des coefficients de similarité des arcs incidents (*cases_1*), elle

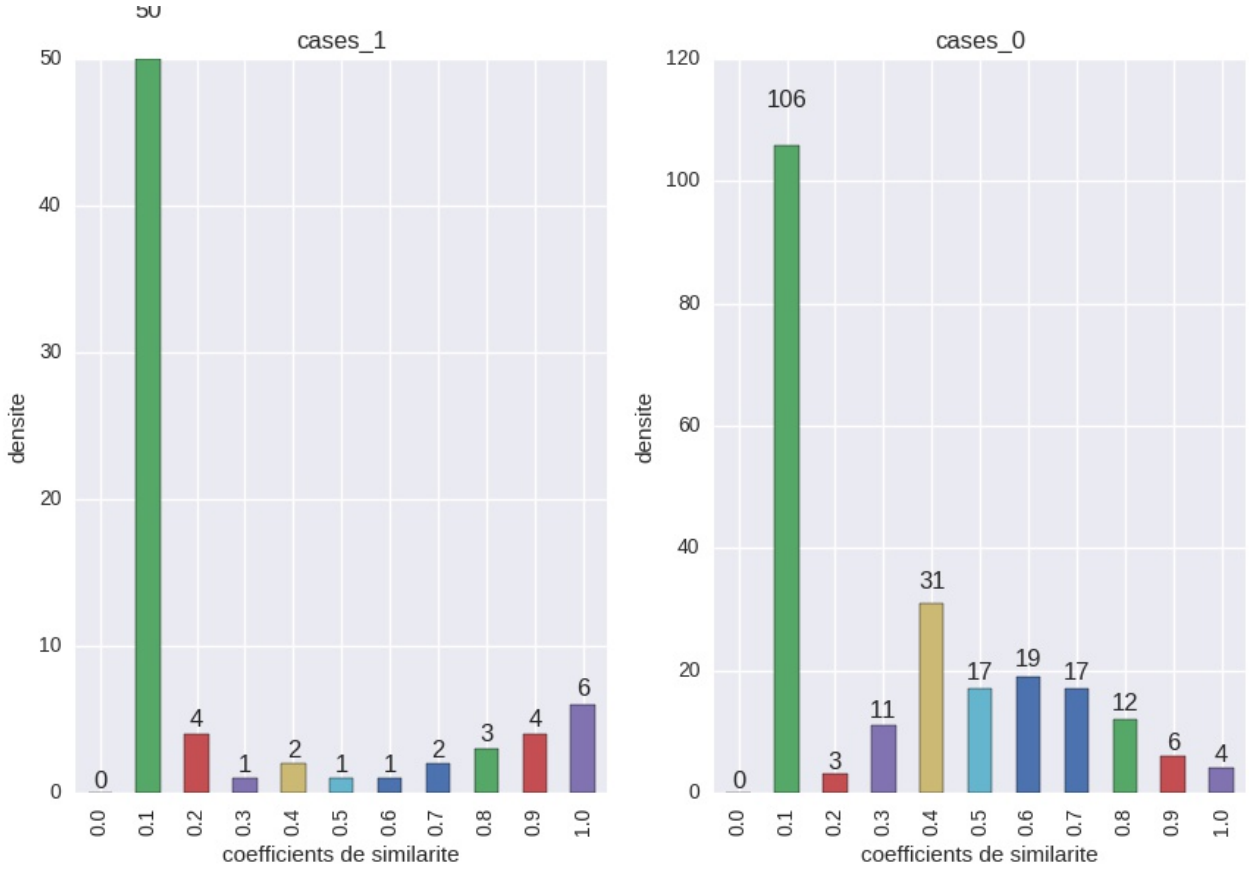

Figure 1.5: distribution des coefficients des similarités selon les arcs qui sont incidents. *cases_1* désigne les arcs incidents et *cases_0* désigne les arcs non incidents dans le *sous-réseau de Champlan*. Le coefficient de similarité 0.1 indique toutes les valeurs dans l'intervalle $[0.1, 0.2[$.

est aussi asymétrique à gauche avec un pic pour les coefficients appartenant à l'intervalle $[0.1, 0.2[$. Pour comprendre cette distribution, nous représentons les profils de consommation de certaines paires d'arcs incidents ayant leur coefficient de similarité appartenant à l'intervalle $[0.1, 0.2[$ dans la figure 1.7. Dans ces représentations, il y'a toujours une courbe constante et sa présence est due à l'impossibilité de collecter de données à cause d'une panne. Cette série contient alors des valeurs nulles. Il y'a aussi la fourniture d'énergie dans les branches. En effet, la source ne fournit que la quantité d'énergie demandée par un équipement. Les séries des arcs sont différentes et la distance de Pearson est la plus faible ($corr(x- > y, y- > z) = 0.1$). Dans ce cas, les erreurs sont introduites par les données et le fonctionnement du réseau.
Dans les cas d'erreurs de similarité énoncées plus haut, nous ne pouvons pas les éviter pendant le calcul des coefficients parce que le mécanisme de récupération des données est défaillant et l'arrêt de la consommation d'électricité d'une branche est masqué par la mise en service de plusieurs serveurs dans une baie appartenant à une autre branche.

Les coefficients de similarité sont regroupées dans une matrice symétrique carrée de dimension $N \times N$ avec N le nombre d'arcs dans le *sous-réseau de Champlan*. Les lignes et les colonnes de la matrice sont les arcs du sous-réseau. Chaque case de la matrice contient un coefficient de similarité entre deux arcs. Les cases de la diagonale de la matrice contiennent les coefficients d'un arc avec lui-même et sont égales à 0. Cette matrice est appelée *matrice de corrélation* et se note M_c. Sur

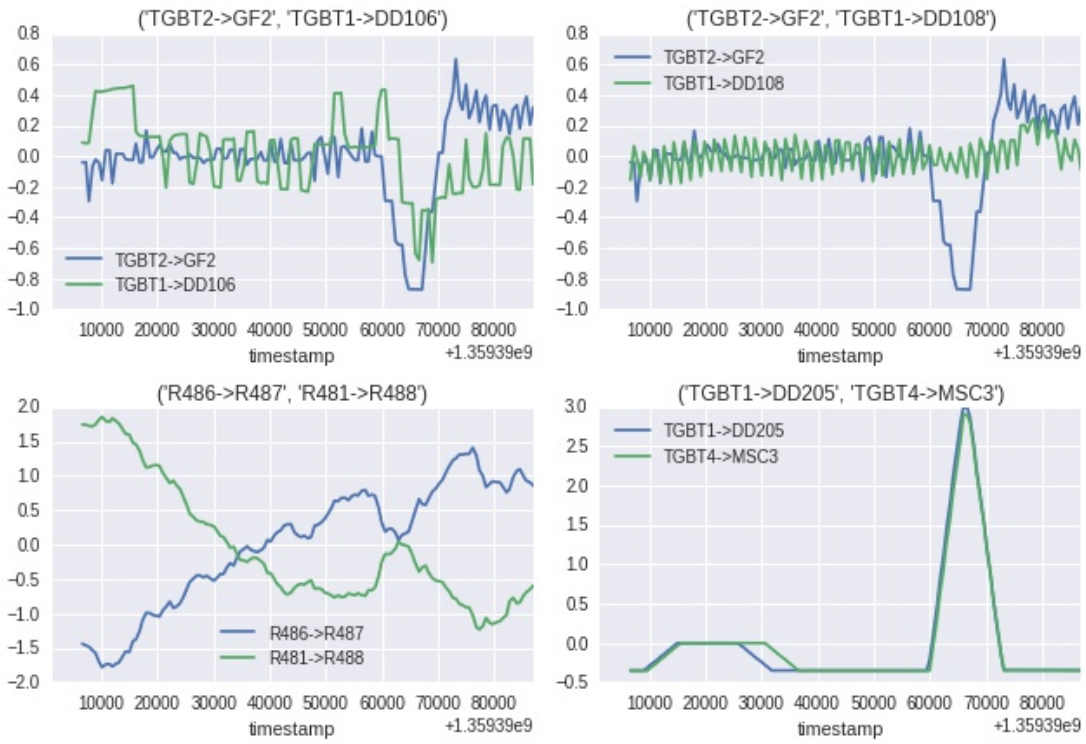

Figure 1.6: Profils de consommation des paires d'arcs n'ayant aucun équipement en commun. En haut à gauche, nous avons les courbes des arcs $TGBT2- > GF2$, $TGBT1- > DD106$. En haut à droite, les courbes des arcs $TGBT2- > GF2$, $TGBT1- > DD108$. En bas à gauche, les courbes des arcs $R486- > R487$, $R481- > R488$. En bas à droite, les courbes des arcs $TGBT1- > DD205$, $TGBT4- > MSC3$

cette matrice, différentes valeurs de seuils $s \in [0, 1]$ sont testées dans le but de déterminer la matrice d'adjacence $M_{c,s}$ du *sous-réseau de Champlan*. Des arcs $x- > y$ et $x- > z$ dont leur coefficient $corr(x- > y, x- > z) \geq s$ ont leur case $M_{c,s}[x- > y, x- > z] = 1$ sinon $M_{c,s}[x- > y, x- > z] = 0$ si $corr(x- > y, x- > z) < s$. La figure 1.8 présente les distributions des relations d'incidences entre les arcs après l'application des seuils sur la matrice de corrélation. Pour un seuil donné, les relations d'incidences sont regroupées en 4 catégories :

- Les incidences *vrais positives* : il existe un sommet en commun entre les arcs et la case associée de la paire d'arcs $x- > y, x- > z$ $M_{c,s}(x- > y, x- > z) = 1$.
- Les incidences *vrais négatives* : il existe aucun sommet en commun entre les arcs et la case associée de la paire d'arcs $x- > y, x- > z$ $M_{c,s}(x- > y, x- > z) = 0$.
- Les incidences *fausses positives* : il existe aucun sommet en commun entre les arcs et la case associée de la paire d'arcs $x- > y, x- > z$ $M_{c,s}(x- > y, x- > z) = 1$.
- Les incidences *fausses négatives* : il existe un sommet en commun entre les arcs et la case associée de la paire d'arcs $x- > y, x- > z$ $M_{c,s}(x- > y, x- > z) = 0$.

Les incidences *fausses positives* et *fausses négatives* constituent des *erreurs d'incidences* dans la matrice d'adjacence $M_{c,s}$. Nous considérons aussi certaines incidences *vrai positives* et *vrais négatives* comme les *erreurs d'incidences*. Chaque graphique affiche le nombre d'erreurs d'incidences associé

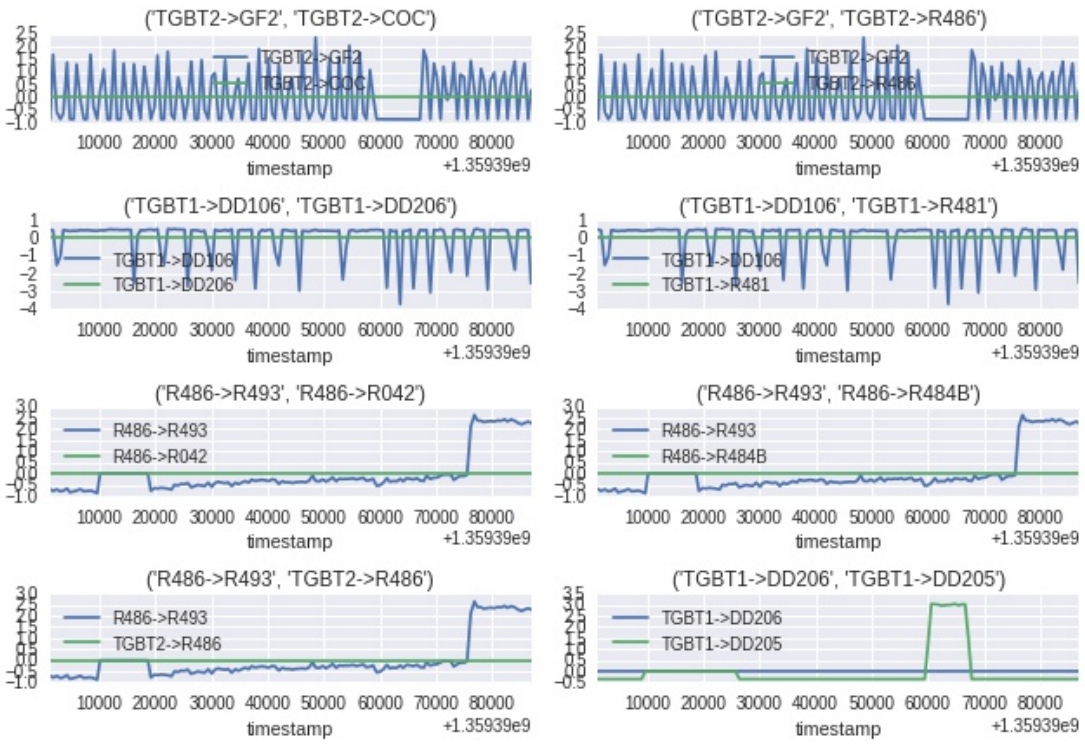

Figure 1.7: Profils de consommation des paires d'arcs ayant un équipement en commun. les arcs $TGBT2- > GF2$ et $TGBT2- > COC$ partagent l'équipement $TGBT2$.

à un seuil. Par exemple, dans le graphique *VraiPositive_ErreurAdjacence* de la figure 1.8 nous avons 17 paires d'arcs partageant un sommet pour un seuil $s = 0.4$. De même, nous avons 57 paires d'arcs n'ayant aucun sommet en commun pour un seuil $s = 0.4$ dans le graphique *fauxNegatives_ErreurAdjacence* de la figure 1.8. Nous observons qu'il n'existe aucun seuil qui maximise les nombres d'incidences *vrai positives* et *vrai négatives* et qui minimise les nombres d'incidences *fausses positives* et *fausses négatives*. Et cela est due à l'introduction des erreurs des coefficients de similarité dans la *matrice de corrélation matE*.

Conclusion : dans cette section, nous avons limité notre étude sur un *sous-réseau de Champlan* dans lequel les équipements ne sont pas alimentés par un onduleur. Ce choix fut préféré à cause de notre hypothèse qui stipule que les variations d'électricité se propagent dans le réseau. Nous avons calculé les coefficients de similarité *corr* avec la grandeur P parce que c'est la seule grandeur qui fournit des mesures en monophasé et en triphasée. Certains coefficients de similarité sont érronnés à cause des données, de la méthode de calcul et du mécanisme de fonctionnement du réseau de Champlan. Ces coefficients forment la *matrice de corrélation* M_c carrée et symétrique dans laquelle sont testés des seuils. Si M_c ne contient aucune erreur de similarité alors il existe un seuil qui détermine la matrice d'adjacence de la topologie du sous-réseau de Champlan. Malheureusement, nous ne sommes pas parvenu à determiner la bonne valeur de seuil et aussi à obtenir les coefficients de similarité qui reflètent les relations d'adjacences des arcs.

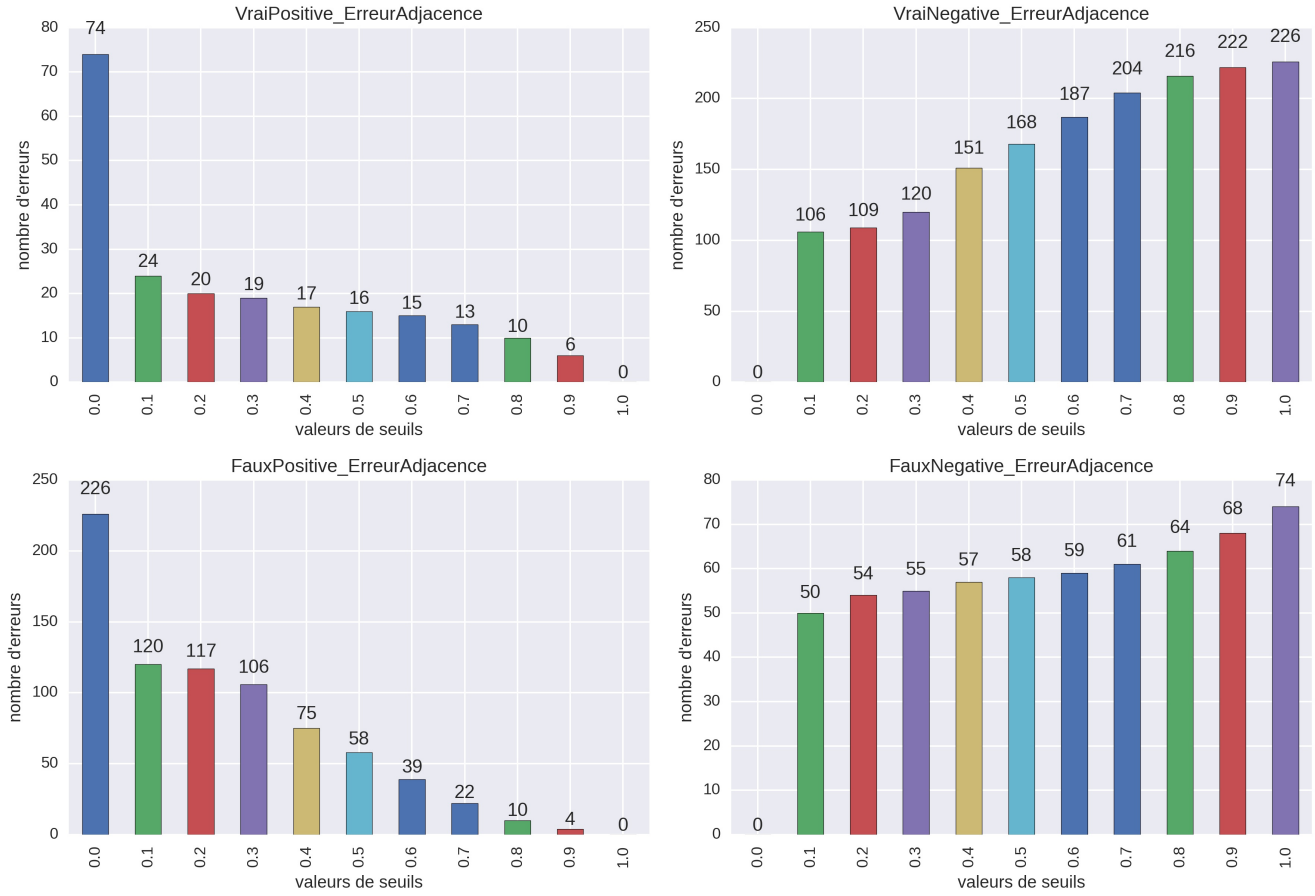

Figure 1.8: distributions des relations d'incidences entre les arcs après l'application de seuils. On distingue 4 relations d'incidences entre les arcs : incidences *fausses positives* (graphique en bas à gauche),*fausses négatives* (graphique en bas à droite), *vrais positives* (graphique en haut à gauche) et *vrai négatives* (graphique en haut à droite).

1.4 Conclusion du chapitre

Ce chapitre se subdivise en quatres parties. La première partie présente les domaines dans lesquels l'analyse des séries temporelles est importante. Ensuite, nous exposons notre problème qui consiste à comparer deux séries temporelles en supposant que les variations dans une série sont reproduites dans l'autre série. Pour résoudre notre problème, nous décidons de nous servir des méthodes de classification de séries temporelles. Dans la seconde partie, nous détaillons les méthodes qui se regroupent en trois catégories : *similarité sur les séries entières, similarité sur les parties significatives, similarité par aggrégation des caractéristiques descriptives*. Chaque catégorie décrit des distances de similarité. Les avantages et les inconvienents de chaque catégorie sont décrites dans la troisième partie. Ainsi, en analysant nos données et en se basant sur notre hypothèse, nous avons montré que les méthodes sur les séries entières sont adaptées à notre problème. En effet, notre hypothèse stipule que deux arcs partageant un équipement ont les profils de consommation sont similaires et leur coefficient de similarité est proche de 1. Dans le cas contraire, leur coefficient de similarité tend vers 0 et les profils de consommation ont des courbes différentes. Nous avons alors choisi la *distance de Pearson* comme méthode de similarité parce qu'elle est une métrique, de complexité linéaire, ne traite pas le décalage temporel et enfin retourne des valeurs appartenant à

l'intervalle $[0, 1]$. Nous avons appliqué cette distance sur un *sous-réseau du datacenter Champlan* parce que ce sous-réseau ne possède aucun équipement alimenté par un onduleur. Ensuite, la seule grandeur qui contient des valeurs en monophasé et triphasé est la grandeur P. Les coefficients de similarité entre les arcs obtenus avec la grandeur P contiennent des erreurs de similarité. Une erreur de similarité est un coefficient proche de 1 alors que les arcs ne concourent pas en un équipement et vice-versa. Ces coefficients forment la *matrice de corrélation* qui appliquée à un seuil propose la matrice d'adjacence du sous-réseau de champlan. Ceci est vérifié à condition que les coefficients ne contiennent aucune erreur. Enfin, nous avons montré qu'il est difficile de déterminer la bonne valeur de seuil en présence des coefficients de similarité érronnés.