Structuration de l'espace latent d'un auto-encodeur génératif pour la séparation de courbes de charge

Khalid OUBLAL[*1,2] Said LADJAL[1] David BENHAIEM[2] Emmanuel LE BORGNE[2] François ROUEFF[1]

[1]LTCI, Telecom de Paris - Institute Polytechnique de Paris*, Palaiseau, France

[2]R&D One Tech TotalEnergies, 2 Place Jean Millier, 92400 Courbevoie, France

Résumé – Récemment, on a assisté à une explosion de la littérature sur les approches de séparation de courbes de charge basées sur l'apprentissage profond. Toutefois, deux obstacles majeurs persistent : garantir la généralisation du modèle et son interprétabilité. Pour répondre à ces enjeux, notre approche **S-VAE** utilise un modèle d'apprentissage génératif basé sur un autoencodeur variationnel, qui permet de générer des données et de rendre le modèle interprétable grâce à la représentation de son espace latent. Cette méthode permet de capturer les caractéristiques primordiales de la séquence de puissance mixte et de la séparer en plusieurs séquences puissances d'appareils.

Abstract – There has been a surge of literature focusing on deep learning techniques for Energy Disaggregation as a source separation problems. Despite this, the model's ability to generalize and its interpretability are still two major challenges. To tackle these issues, we employ a generative learning model utilizing Structured Variational Autoencoder (**S-VAE**). This approach facilitates the reliability of the model against signature variation and enables the interpretability of the model by designing its latent space in a specific way. Using this method, we can capture the fundamental features of the mixed power sequence, and separate it into distinct power sequences of appliances.

1 Introduction

L'apprentissage profond est de nos jours largement utilisé pour la séparation des courbes de charge d'énergie, en raison de sa capacité à apprendre des motifs complexes dans les données. Cependant, ces approches souffrent souvent de problèmes d'interprétabilité, ce qui rend difficile la compréhension des décisions prises par le modèle. De plus, la généralisation à de nouveaux contextes reste un défi important. Afin de répondre à ces préoccupations, plusieurs approches ont été proposées, telles que l'utilisation de réseaux de neurones convolutionnels (CNN) pour extraire des caractéristiques des formes de puissance consommée, proposée par Ciancetta et al. [3]. Bien que cette approche ait donné des résultats prometteurs sur le jeu de données UK-DALE [4], elle pose un problème de généralisation. Chen et al. [1] ont proposé une approche de séquence en séquence (S2S) combinant les CNN et LSTM, qui ont atteint des performances remarquables sur l'ensemble de données REDD [6]. Les auto-encodeurs de débruitage (DAE) ont également été introduits pour désagréger la puissance d'un équipement à partir d'une courbe de charge bruitée [8].

Les réseaux de neurones récurrents (RNN) ont également été utilisés dans le contexte de la désagrégation de la consommation d'énergie pour modéliser les dépendances temporelles dans les données. Yang et al. [9] ont proposé une nouvelle approche basée sur les RNN, appelée S2P, qui utilise des unités récurrentes (GRU) et des mécanismes d'attention pour modéliser les relations complexes entre les appareils, atteignant des performances de pointe sur l'ensemble de données UK-DALE. Cependant, comme pour les autres approches basées sur l'apprentissage profond, le manque d'interprétabilité et de généralisabilité reste une préoccupation majeure.

Ce travail est financé par One Tech. Un brevet a été déposé par les auteurs et octroyé à TotalEnergies SE.

Ce travaille cherche à répondre à deux questions : **l'utilisation d'un modèle génératif garantit-il une séparation généralisable ? Comment apprendre une représentation interprétable de l'espace latent ?**

2 Formulation du problème

Soit $X_t \in \mathbb{R}^c$ la puissance agrégée mesurée bruitée pour l'ensemble du foyer analysé à un instant t, et avec $c = 3$ (correspondant aux puissances active, réactive et apparente)[1]. On note $x_t \in \mathbb{R}$ la puissance active, qui s'écrit comme la somme des contributions de chaque appareil $y_{t,m}$, $m = 1, \dots, M$, et d'un bruit résiduel ξ_t :

$$x_t = \sum_{m=1}^{M} y_{t,m} + \xi_t \tag{1}$$

L'indice m fait référence au m-ième appareil électrique parmi les M disponibles. Le problème consiste donc à déduire, à partir d'une séquence $X^{(t)} := X_{t:t+\tau}$ de longueur $\tau + 1$, les composantes correspondantes $Y^{(t)} := y^{(t)}_{1:M} := y_{t:t+\tau,1:M}$.

On note $\mathcal{D} = \{X^{(t)}, Y^{(t)}\}_{t=1}^{N}$ l'ensemble d'apprentissage, où $X^{(t)} \in \mathbb{R}^{c\times(\tau+1)}$ et $Y^{(t)} \in \mathbb{R}^{M\times(\tau+1)}$. Les séquences $X^{(t)}$ représentent un ensemble d'échantillons provenant d'une distribution inconnue. L'autoencodeur variationnel (VAE) vise à inférer cette distribution avec un modèle paramétrique comportant une variable latente (non-observée). On définit cette variable latente sous la forme d'une variable $Z^{(t)}$ de dimension

[1]La puissance électrique se compose de trois types : active, réactive et apparente. La puissance active, est la quantité d'énergie électrique réellement convertie en travail utile. La puissance réactive, est la quantité d'énergie électrique stockée temporairement et échangée entre la source d'alimentation et le dispositif électrique, sans produire de travail utile. La puissance apparente, est la somme vectorielle de la puissance active et de la puissance réactive.

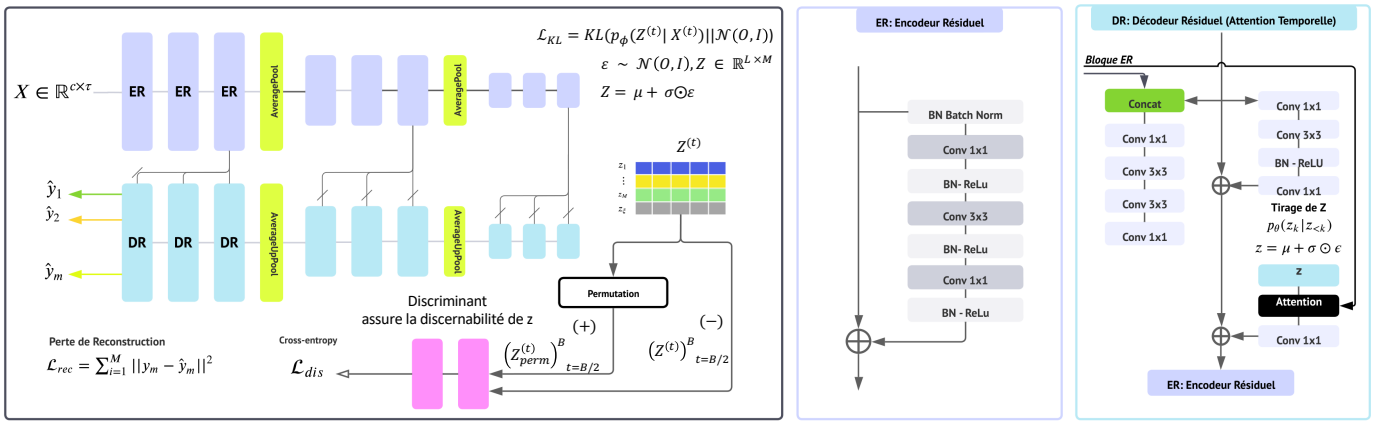

FIGURE 1 : A gauche, illustration du processus d'apprentissage de **S-VAE**. Sur la droite, la description de l'encodeur résiduel **(ER)** et du décodeur résiduel **(DR)**, y compris l'unité d'attention temporelle **«Attention»**.

$(M+1) \times L$ «représentant» ou «codant» une séquence multivariée $X^{(t)}$. Notre objectif est de proposer une architecture de VAE qui permet d'imposer une structure de la variable latente en lien avec les activations de chaque équipement électroménager.

3 Méthodes proposées

Notre approche de la désagrégation de la courbe de charge vise à identifier et à séparer avec précision les contributions des différents équipements électroménagers dans une séquence de puissance agrégée donnée $x_{t:t+\tau}$.

Soient (f_ϕ, f_θ) un couple encodeur/décodeur. Cela signifie que, étant donné le code latent $Z^{(t)}$, $X^{(t)}$ suit une loi paramétrée par $f_\theta(Z^{(t)})$. De son côté, le code latent $Z^{(t)}$, étant donné $X^{(t)}$, suit une loi approchée par une loi paramétrée par $f_\phi(X^{(t)})$.

Dans le contexte de la désagrégation énergétique, comme on s'intéresse à la décomposition de la puissance active $x^{(t)} = x_{t:t+\tau}$ à partir des composantes de la séquence $Y^{(t)}$, supposées indépendantes, l'encodeur cherche directement à décrire la loi des $y_m^{(t)}$ pour $m = 1 : M$. Nous cherchons de plus à factoriser le code latent sous forme : $Z = z_{1:M+1}$ où $z_m \in \mathbb{R}^L$ représente le code latent de l'appareil $m = 1 : M$, tandis que $z_{M+1} \in \mathbb{R}^L$ représente le code latent du résidu ξ. La dimension L correspond alors à la dimension nécessaire pour coder la signature du signal de chaque équipement le long d'une séquence de taille $\tau + 1$.

3.1 Fonction de perte pour la désagrégation d'énergie

L'approximation de la loi conditionnelle de z est donnée par une famille variationnelle gaussienne $q_\phi(Z^{(t)}|X^{(t)}) = \mathcal{N}(Z^{(t)}; \mu(X^{(t)}, \phi), \text{diag}(\sigma^2(X^{(t)}, \phi)))$, où $\mu(X^{(t)}, \phi)$ et $\sigma^2(X^{(t)}, \phi)$ sont les sorties de l'encodeur (voir **ER** dans la Figure.1) qui paramètrent la moyenne et la variance de la distribution gaussienne, respectivement. Comme dans [2], le VAE est obtenu en optimisant la borne inférieure d'«évidence» (dite ELBO) qui contient une terme de divergence de KL et un terme de reconstruction. Dans notre approche, le terme KL est identique : $\text{KL}(q_\phi(Z^{(t)}|X^{(t)})||p(Z^{(t)}))$, avec p désignant la loi a priori de $Z^{(t)}$ prise gaussienne centrée et normalisée, ce qui mène au terme

$$\frac{1}{2}\sum_{j=1}^{J}\left(1 + \log\sigma_j^2(X^{(t)}, \phi) - \mu_j^2(X^{(t)}, \phi) - \sigma_j^2(X^{(t)}, \phi)\right)$$

où $J = (M+1)\,L$. En revanche le terme de reconstruction est différent et s''écrit :

$$\sum_{t}^{t+\tau}\sum_{m=1}^{M}\|y_{t,m} - \hat{y}_{t,m}\|^2 \qquad (2)$$

où $\hat{y}_{t,m}$ désigne la puissance prédite à l'instant t par la sortie m-ème du décodeur f_θ appliqué à un $Z^{(t)}$ simulé sous le paramètre ϕ.

Dans la suite, la somme en $t = 1, \ldots, N$ sur l'ensemble d'apprentissage des termes KL et des erreurs de reconstruction serons écrites : $\mathcal{L}_{\text{KL}}$ et $\mathcal{L}_{rec}$, respectivement.

3.2 Structuration de l'espace latent

La séparation des distributions multidimensionnelles en composantes statistiquement indépendantes et interprétables est un problème complexe en apprentissage profond. Plusieurs méthodes ont été proposées pour résoudre ce problème comme β-TCVAE [2]. La méthode la plus couramment utilisée est la «Total Correlation», qui introduit la pénalité TC définie par la divergence $\text{KL}(p_\phi(Z)||\prod_j p_\phi(z_j))$. Toutefois, l'estimation de cette divergence est à la fois coûteuse et difficile à réaliser.

- **Estimation de TC.** Comme dans [2], on utilise un discriminateur D_ψ qui est chargé de remplacer la pénalité, notée TC dans la suite, en optimisant la performance d'un discriminateur entre la distribution de variable latentes et d'une version permutée de celle-ci. Le discriminateur D_ψ et le modèle sont entraînés simultanément (voir Algorithm.1). Pour son apprentissage, les variables latentes de la moitié du batch sont permutées aléatoirement, créant ainsi des cas positifs Z_{perm} dont les composantes sont indépendantes, et l'autre moitié est laissée telle quelle, correspondant aux cas négatifs où les composantes de Z sont possiblement liées.

- **Correspondance entre $z_j^{(t)}$ et $y_j^{(t)}$.** L'objectif de ce processus est de faire correspondre la composante z_j avec

l'appareil j en utilisant une stratégie de masquage durant l'apprentissage. Nous voulons forcer le modèle à reconstruire les signaux $y_j^{(t)}$ en ne tenant compte que de $z_j^{(t)}$. Pour ce faire, durant certaines étapes d'apprentissage au lieu de donner en entrée une séquence normale $X^{(t)}$ nous donnons en entrée une séquence $y_j^{(t)}$ correspondant à un appareil j (notez que nous possédons les trois valeurs active, réactive et apparente pour les appareils). La seule sortie évaluée est celle correpondant à l'appareil j (comme si l'appareil était seul). Et l'espace latent est modifiée en masquant les $z_{m\neq j}^{(t)}$ (on les tire aléatoirement. La composante $z_j^{(t)}$ n'est pas altérée. Cela force le réseau à savoir déduire $y_j^{(t)}$ avec comme seule valeur utile la composante $z_j^{(t)}$. En pratique pour chaque batch de taille 256, les 50 dernières entrées sont des exemples pour lesqules les puissances active, réactive et apparente d'un seul appareil sont mises en entrée et la fonction de perte n'est évaluée que sur y_j.

Les trois termes, reconstruction, divergence de KL et TC, sont pénalisés par deux poids d'importance relative α, β. La fonction de coût minimisée est défini comme ci-dessous :

$$\mathcal{L}(\alpha, \beta; \theta, \phi, \psi) = \mathcal{L}_{rec} + \alpha\, \mathcal{L}_{\mathrm{KL}} + \beta\, \mathrm{TC} \qquad (3)$$

Algorithme 1 : Apprentissage du modèle avec une contrainte de structuration

Observations $(X^{(t)})_{t=1}^{N}$, $(Y^{(t)})_{t=1}^{N}$, taille du batch B, α,β, η, optimiseur Adam [5], taille maximale de l'espace latent $(M+1) \times L$, avec M est le nombre d'appareils a trouver. Initialisation des paramètres θ, ϕ, ψ de l'encodeur f_θ, décodeur f_ϕ et le discriminateur D_ψ respectivement.

tant que *la fonction de perte $\mathcal{L}_{\theta,\phi,\psi}$ n'a pas convergé* **faire**

- Sélectionner un batch $(X^{(t)})_{t\in\mathcal{B}}$ de taille B
- Tirage $Z_\theta^{(t)} \sim q_\phi(Z^{(t)}|X^{(t)})$ pour tous les $t \in \mathcal{B}$
- $\mathcal{L}_{\theta,\phi,\psi} \leftarrow \mathcal{L}(\alpha, \beta; \theta, \phi, \psi)$ Eq.(3)
- Claculer la fonction de perte, et gradient de D_ψ :
- $\psi \leftarrow \psi + \frac{\eta}{2B} \nabla_\psi [\sum_{t\in \frac{\mathcal{B}}{2}} \log D_\psi(Z_{\mathrm{perm}}^{(t)})$
- $+\sum_{t\in \frac{\mathcal{B}}{2}} \log(1 - D_\psi(Z^{(t)}))]$
- Calculer le gradient $\nabla_{\theta,\phi}\mathcal{L}(\theta)$

fin

3.3 Attention Temporelle sur l'espace latent

Afin de tirer avantage corrélation temporelle entre les séquences $X^{(t)}$ et $X^{(t+1)}$ nous ajoutons une couche d'attention qui utilise les résultats des calculs pour le temps t afin de les utiliser pour le temps $t+1$. Cette modification a deux conséquences. D'une part, notre réseau ne sépare plus strictement les séquences $X^{(t)}$ et $X^{(t+1)}$ et a donc besoin de séquences plus longues pour donner des décisions de bonne qualité (typiquement des séquences de longueur double de τ). D'autre part, au moment du test il faut fournir au réseau un batch constitué de séquences successives. Nos expériences avec des batch de séquences non successives (et une séparation stricte des éléments du batch) étaient beaucoup moins satisfaisants.

4 Expériences et Résultats

4.1 Données

Nous avons mené des expériences sur deux ensembles de données accessibles au public, à savoir UK-DALE [4] et REDD [6]. L'ensemble de données UK-DALE [4] se compose de 5 logements avec un nombre variable d'appareils dotés de compteurs divisionnaires et comprend des mesures de puissance agrégées et individuelles au niveau de l'equipement, échantillonnées à 1/6 Hz. Nous avons concentré notre analyse sur trois équipements spécifiques : Réfrigérateur, Machine à Laver **(MaL)**, et Four. De même pour REDD [6] on récupère l'ensemble des 6 logements. Afin d'évaluer la capacité de généralisation des modèles, on a entraîné les modèles sur le jeu de données **REDD**[6] et on les a testés sur le jeu de données **UK-DALE**[4], puis on a inversé cette procédure.

4.2 Encodeurs f_ϕ et Décodeurs f_θ

Notre modèle (Figure1) utilise un encodeur bidirectionnel [7], qui traite les données d'entrée de manière hiérarchique pour produire un code latent à faible résolution qui est affiné par une série de couches de suréchantillonnage. La première phase consiste en un encodeur rudimentaire qui produit un code latent de basse résolution. Ce code est ensuite affiné par une série de couches de sur-échantillonnage dans les blocs «**Décodeurs Résiduels**», ce qui augmente progressivement la résolution. Pour chaque étape du processus de raffinement. L'utilisation des **Encodeurs et Décodeurs Résiduels** nous permet efficacement de capturer les caractéristiques sémantiques, tandis que l'attention temporelle dans les **Décodeurs Résiduels** implémenter par GRU assure la dépendance temporelle de z.

Dans notre architecture, la plus petite dimension de z est fixé à $z \in \mathbb{R}^{(M+1)\times L}$ avec $L = 16$ et $M = 3$, c'est le nombre d'équipements à séparé dans une séquence mixte de taille $\tau = 256$.

4.3 Optimisation

Lors de toutes nos expériences, nous avons utilisé l'optimiseur Adam [5] avec un taux d'apprentissage initial de 10^{-3} et une décroissance cosinus du taux d'apprentissage. Nous avons également réduit le taux d'apprentissage à 7×10^{-4} pour augmenter la stabilité de l'entraînement et appliqué un arrêt précoce après 5 itérations. On fixe $\alpha = 0.5$ et $\beta = 2.5$ après une recherche en grille sur la meilleure convergence du modèle sur les données de validation.

4.4 Résultats

Les résultats sont présentés dans la Table.(1). Le critère MSE correspond au critère d'attache aux données moyenné sur la base de test. On utilise un seuillage identique à [8] pour détecter la présence d'un équipement donné. Ce problème de détection est évalué par le score **F1** sur la base de test. Notre approche dépasse S2S [1], DAE [8], S2P [9] en termes de **MSE** et **F1**. La correspondance entre z_j et y_j est illustrée par la figure (2). Les vecteurs z_j ont été projeté sur les deux dimensions les plus importantes de leur ACP. Les points sont colorés suivant que la machine est active ou inactive. Une ligne i correspond l'activation ou non de la machine i (identifiée

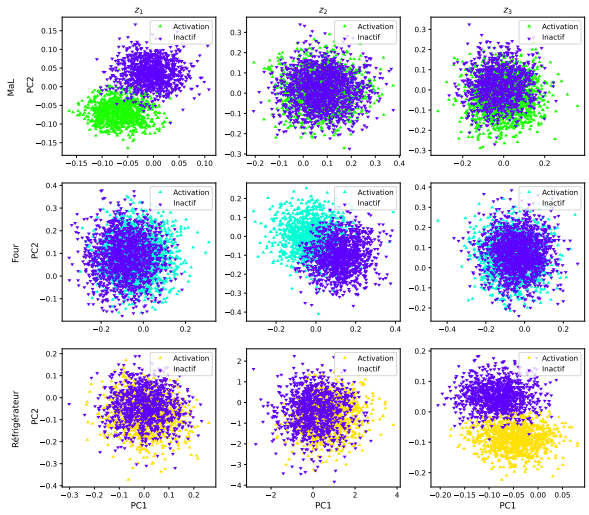

FIGURE 2 : Chaque ligne correspond à des données où au plus un équipement (Machine à Laver «MaL», Four et Réfrigérateur de haut en bas) est actif. On représente dans chaque colonne la composante $z_m^{(t)}$ de la variable latente $Z^{(t)}$ structurée lors de l'apprentissage pour correspondre à l'activation de l'équipement m. On représente uniquement les 2 premières composantes ACP de $z_m^{(t)}$ de couleurs différentes suivant que l'équipement est activé ou non (couleur bleu) dans l'échantillon considéré. Les graphiques diagonaux montrent que $Z^{(t)}$ est structuré en fonction de l'activation de l'équipement m.

par la couleur) et une colonne à la visualisation de z_j. On remarque que sur la diagonale les points de différentes couleurs (active/non active) sont correctement séparés par une droite. Cela conforte notre procédure de mise en correspondance entre z_j et y_j. Cela permet une meilleures explicabilité des résultats et d'investiguer les cas d'échec éventuels.

TABLE 1 : Résultats sur les données **UK-DALE** et **REDD** : erreur quadratique moyenne (MSE) en $Watt^2$ et score $F1$ calculés sur les données de test.

Dataset Test on	Method	Metric	Réfrigérateur	MaL	Four
UK-DALE [4]	DAE [8]	F1 (↑)	80.57	81.37	81.80
	S2S [1]		83.99	86.08	83.61
	S2P [9]		83.73	86.12	83.63
	S-VAE		**91.81**	**93.26**	**93.77**
	DAE [8]	MSE (↓)	25.74	25.63	25.46
	S2S [1]		26.70	24.72	23.98
	S2P [9]		27.36	28.92	25.28
	S-VAE		**19.55**	**18.33**	**19.30**
REDD [6]	DAE [8]	F1 (↑)	82.99	81.94	81.90
	S2S [1]		87.09	86.16	83.78
	S2P [9]		86.96	85.57	84.14
	S-VAE		**94.25**	**93.07**	**94.04**
	DAE [8]	MSE (↓)	26.56	25.34	25.42
	S2S [1]		26.56	24.78	23.94
	S2P [9]		30.68	28.40	25.04
	S-VAE		**19.48**	**18.33**	**19.55**

5 Conclusion et Perspectives

Dans cet article, nous avons proposé une approche interprétable pour la désagrégation de courbes de charge. Le modèle proposé utilise une architecture codeur-décodeur similaires à un VAE et vise à améliorer la structuration de l'espace latent. Cette approche se révèle plus performante que les techniques récentes utilisées dans l'état de l'art. Elle permet de plus de visualiser l'état d'activation des appareils en structurant l'espace latent au cours de l'apprentissage. Dans de futurs travaux, nous envisageons d'analyser l'espace latent de ce modèle en conjonction avec des caractéristiques plus complexes que la simple activation pour chaque appareil et déterminer les conditions de cas d'échec pour savoir s'ils sont prévisibles à partir des latents.

Références

[1] Kunjin CHEN, Qin WANG, Ziyu HE, Kunlong CHEN, Jun HU et Jinliang HE : Convolutional sequence to sequence non-intrusive load monitoring. *the Journal of Engineering*, 2018(17):1860–1864, 2018.

[2] Ricky T. Q. CHEN, Xuechen LI, Roger B GROSSE et David K DUVENAUD : Isolating sources of disentanglement in variational autoencoders. *In* S. BENGIO, H. WALLACH, H. LAROCHELLE, K. GRAUMAN, N. CESA-BIANCHI et R. GARNETT, éditeurs : *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[3] Fabrizio CIANCETTA, Giovanni BUCCI, Edoardo FIORUCCI, Simone MARI et Andrea FIORAVANTI : A new convolutional neural network-based system for nilm applications. *IEEE Transactions on Instrumentation and Measurement*, 2021.

[4] Jack KELLY et William KNOTTENBELT : The uk-dale dataset, domestic appliance-level electricity demand and whole-house demand from five uk homes. *Scientific data*, 2, 2015.

[5] Diederik P KINGMA et Jimmy BA : Adam : A method for stochastic optimization. *arXiv preprint arXiv :1412.6980*, 2014.

[6] J Zico KOLTER et Matthew J JOHNSON : Redd : A public data set for energy disaggregation research. *In Workshop on data mining applications in sustainability (SIGKDD), San Diego, CA*, volume 25, 2011.

[7] Arash VAHDAT et Jan KAUTZ : Nvae : A deep hierarchical variational autoencoder. *In* H. LAROCHELLE, M. RANZATO, R. HADSELL, M.F. BALCAN et H. LIN, éditeurs : *Advances in Neural Information Processing Systems*, volume 33, pages 19667–19679. Curran Associates, Inc., 2020.

[8] Michele VALENTI, Roberto BONFIGLI, Emanuele PRINCIPI et Stefano SQUARTINI : Exploiting the reactive power in deep neural models for non-intrusive load monitoring. *In 2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018.

[9] Mingzhi YANG, Xinchun LI et Yue LIU : Sequence to point learning based on an attention neural network for nonintrusive load decomposition. *Electronics*, (14), 2021.