

Inférence Grammaticale pour la Prédiction de la Consommation Énergétique

G. Guerard¹, H. Pousseur², M. Rivoire²

¹ Pole Universitaire Léonard de Vinci, Research Center, DVRC

² Research Student, DVRC

92 916, Paris La Défense, France

prenom.nom@devinci.fr

Résumé

La domotique est programmable, contrôlable, et la plupart du temps a une consommation connue. Néanmoins, la prévision d'une consommation des ménages représente un problème difficile généralement résolu à l'aide de l'apprentissage en profondeur ou de modèles mathématiques coûteux en temps et en mémoire. L'approche proposée est basée sur l'inférence grammaticale pour prédire la consommation d'un appareil à partir du passé. L'automate de prédiction consomme peu de ressources et peut être facilement intégré dans des appareils intelligents, des compteurs intelligents ou des maisons intelligentes.

Mots-clés

Prédiction, Inférence grammaticale, ALERGIA

Abstract

Home automation is programmable, schedulable, and most of the time has a known consumption. Nevertheless, forecasting a household consumption represents a challenging problem typically resolved using deep learning or huge mathematical models. The proposed approach is based on grammatical inference to predict a device's consumption from the past. The prediction machine consumes little resources and can be easily embedded into smart devices, smart meters or smart houses.

Keywords

Forecast, Grammar Induction, ALERGIA

1 Introduction

Au cours de la dernière décennie, les applications de prévision énergétique ont été développées non seulement du côté réseau des réseaux électriques, mais aussi du côté client afin d'équilibrer la charge et de la demande. Dans ce contexte, la prévision à court terme de la consommation d'énergie électrique est une condition nécessaire à la gestion de l'énergie et à la planification de tous les bâtiments, des ménages et des résidences à petite échelle aux énormes complexes immobiliers à grande échelle. Elle permet de surveiller la consommation d'énergie et de trouver les creux et les pics de demande, de réduire les pertes, de

minimiser les risques, de garantir la fiabilité pour un fonctionnement ininterrompu. La prévision des charges joue un rôle actif dans la prise de décisions viables en ce qui concerne la planification de la maintenance et les investissements futurs, y compris les technologies énergétiques renouvelables et non renouvelables.

Plus récemment, l'omniprésence de l'Internet des objets rend les systèmes énergétiques distribués plus intelligents en optimisant l'efficacité énergétique pour réduire les pertes et crée une nouvelle aire appelée Internet de l'énergie (IoE) qui est équipée de systèmes de prévision intelligents. Ces derniers utilisent des prévisions météorologiques et d'autres explications pour prévoir la consommation d'énergie future. Selon l'un des derniers rapports de l'Agence internationale de l'énergie¹, les bâtiments représentent la plus grande partie de la consommation finale d'énergie avec une part de 36% du marché mondial, ce qui augmente l'importance des prévisions énergétiques des bâtiments afin de rétablir l'équilibre entre l'offre et la demande pour un avenir plus économe en énergie.

Il n'existe actuellement pas de norme définissant les types de prévisions. Hong et Fan ont regroupé les catégories de prévisions en très court terme, court terme, moyen terme et long terme avec des horizons de coupure de 1 jour, 2 semaines et 3 ans [9]. Principalement, les prévisions à court terme se réfèrent à des prévisions à l'heure, à la journée ou à la semaine à venir et il est considéré que ce concept peut également être appliqué à la prévision de la consommation d'énergie électrique des bâtiments [23].

Les algorithmes d'apprentissage automatique de la littérature sont couramment utilisés pour prévoir la consommation d'énergie électrique des bâtiments à court terme. La littérature contient une variété d'étude résumant les méthodologies de prévision de la consommation d'énergie des bâtiments sous différents angles.

Zhao et Magoules ont examiné les prévisions de consommation d'énergie des bâtiments en classant les méthodologies telles que les méthodes d'ingénierie, les méthodes statistiques et les méthodes d'IA [22]. Ahmad et al. ont ré-

¹ 2019 Global Status Report for Buildings and Construction : Towards a zero-emission, efficient and resilient buildings and construction sector.

sumé les applications des réseaux de neurones artificiels (ANN) et des machines à vecteurs de support (SVM) [1]. Raza et Khosravi ont mené une étude sur les techniques de prévision de la demande de charge basées sur l'IA, non seulement pour les bâtiments, mais aussi pour les réseaux intelligents, en expliquant toutes les phases de la prévision de charge à court terme [16]. Daut et al. ont examiné la prédiction de la consommation d'énergie électrique des bâtiments en divisant les méthodologies en méthodes conventionnelles, IA et hybrides [5]. Wang et Srinivasan ont comparé des modèles simples et d'ensemble pour la prévision de la consommation d'énergie des bâtiments basée sur l'IA [20].

Wei et al. ont présenté des approches d'apprentissage machine basées sur l'étude des données pour la prévision et la classification de la consommation d'énergie des bâtiments [21]. De la même manière, Amasyali et El-Gohary ont examiné les études de prévision de la consommation d'énergie des bâtiments basées sur les données en se concentrant particulièrement sur les domaines de prédiction, les propriétés des données et les méthodes de prétraitement, les algorithmes d'apprentissage automatique et les mesures de performance [2]. Enfin, Runge et Zmeureanu ont présenté une étude de la prévision de la consommation d'énergie dans les bâtiments utilisant des ANN en mettant en évidence les applications, les données, les modèles de prévision et les mesures de performances [17].

L'apprentissage machine et l'apprentissage profond sont efficaces et proposent des modèles avec des dizaines de colonnes en entrée (en fonction des capteurs / données externes et internes aux bâtiments) pour prédire la consommation énergétique. Cependant, un détail n'est jamais abordé dans ces études : la consommation énergétique d'un tel dispositif par rapport aux gains potentiels.

Notre étude propose une méthode de prédiction peu consommatrice en temps et en mémoire, donc énergétiquement sobre. La méthode se base sur l'inférence grammaticale, c'est-à-dire un apprentissage par construction d'un automate stochastique à partir d'une série temporelle représentant la consommation énergétique d'un appareil.

Le papier est construit comme suit : la deuxième section présente la méthodologie ; cette dernière est décomposée dans les sections 3, 4 et 5 par les traitements sur le jeu de données, les opérations sur ces dernières et la méthode d'inférence grammaticale. La section 6 présente les résultats et le papier se conclut par la section 7.

2 Méthodologie

2.1 Jeu de données

Chaque appareil produit une courbe de consommation. Celui-ci est enregistré par un compteur intelligent sous la forme d'une série temporelle où chaque tuple fournit les informations suivantes : ID_périphérique, consommation (en Wh), date. La courbe de consommation est considérée sans erreur car le compteur intelligent transmet une valeur consolidée. Le système d'éclairage, l'ordinateur et les petits appareils électroménagers ne sont pas pris en compte.

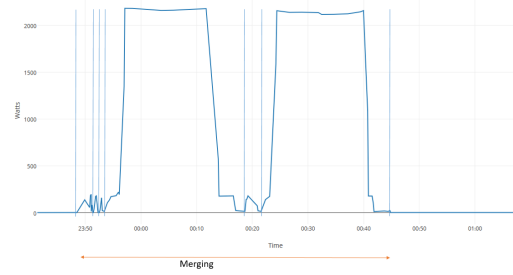


FIGURE 1 – Cycle de consommation d'une machine à laver.

Afin d'apprendre de la consommation d'un appareil, nous devons définir des séquences de consommation. Cette dernière commence lors du passage d'une valeur de consommation nulle à une valeur de consommation non nulle sur deux étapes consécutives de la série temporelle. Dès qu'il y a une valeur nulle de consommation à l'étape suivante, la séquence s'arrête.

Certains appareils peuvent avoir des cycles de consommation programmés avec de courtes périodes à consommation nulle. La méthode suivante fusionne deux séquences de consommation si elle est considérée sur un même cycle de consommation :

$$\Delta B \leq \Delta T_i \text{ and } \Delta B \leq \Delta T_j$$

avec les variables suivantes : ΔB le temps de césure ; ΔT_i le temps de consommation de la séquence i ; ΔT_j pour la séquence j .

La figure 1 présente un exemple de fusion de plusieurs séquences (séparées par des lignes verticales) pour constituer un cycle de consommation unique.

2.2 Méthodes

L'approche proposée dans ce papier est composée de trois étapes distinctes :

- TRAITEMENT DES DONNÉES : tout d'abord, les séquences sont définies comme présentées précédemment. Le bruit est éliminé par une méthode de regroupement.
- PRÉRÉGLER LES DONNÉES : ensuite, les séquences sont analysées en motifs récurrents.
- INFÉRENCE GRAMMATICALE : un automate stochastique est construit par inférence grammaticale des séquences
- PRÉDICTION ET SIMULATION : à partir d'un début de séquence, il est possible de prédire les possibilités futures, mais aussi d'obliger l'automate stochastique à adopter un comportement donné.

3 Traitements des données

La consommation des appareils est considérée en Wh. Étant donné le très grand nombre de symboles que cela produit, il est pertinent de nettoyer les données : un premier algorithme pour discrétiser les données, nommé *Discr. Algo.* ; un deuxième algorithme partitionne l'alphabet, nommé *Clust. Algo.*

3.1 Réduction des données

Discr. Algo. a pour objectif, à partir d'un échantillon de toutes les données de consommation appelé *data*, à fournir un ensemble de données discrétisé à un timer constant qui commence à la valeur de 5 minutes. *Discr. Algo.* calcule entre chaque valeur la pente. En effet, les appareils peuvent avoir des cycles de consommation similaires, mais avec des variations de la puissance requise (par exemple les types distinctifs de cycle de lavage d'une machine à laver).

Une fois que toutes les pentes sont calculées dans l'ensemble de données suivant *données discrétisées*, l'intégrale de la courbe *données* est comparée à l'intégrale de la courbe *données discrétisées*. Le rapport entre ces deux courbes fournit une estimation de la perte d'informations. Ce ratio ne doit pas dépasser une valeur α .

Tant que ce rapport est strictement inférieur à α , il est possible de considérer les données avec un timer plus grand. Au contraire, si le rapport est supérieur à α , il faut raccourcir le timer. L'algorithme s'arrête lorsque le rapport est proche de α .

3.2 Partitionnement des données

Discr. Algo. génère un grand alphabet alors que certaines valeurs de pente sont proches. Pour réduire la taille de l'alphabet et surtout pour réduire le bruit entre les valeurs, des valeurs similaires sont assimilées à une unique. Ce processus est appelé partitionnement.

Le partitionnement représente le processus de détermination de groupes typiques, appelés partitions, dans un ensemble de données. L'objectif est de trouver les partitions les plus homogènes et les plus distincts possible des autres partitions. Plus formellement, le regroupement devrait maximiser la variance inter-partition tout en minimisant la variance intra-partition.

Une classification des méthodes de partitionnement pour diverses données statiques est proposée dans [10]. Dans le contexte de notre ensemble de données, le partitionnement basé sur les centroïdes convient.

Le problème contient une dimension (les valeurs de consommation), donc le centre de la partition est une moyenne arithmétique des valeurs. L'algorithme k-moyennes convient exactement à l'ensemble de données et a une petite complexité de temps et de mémoire. Cet algorithme est linéaire en fonction du k, du nombre de points et du nombre d'itérations.

Comme dans l'algorithme précédent, il ne doit pas perdre trop d'informations. Le rapport entre la différence des courbes intégrales *data clustered* et *data* ne doit pas dépasser une valeur α' .

Tant que ce ratio est strictement inférieur à α' , il est possible de diminuer la valeur k des k-moyennes. Au contraire, si le rapport est supérieur à α' , on augmente la valeur de k . Notez que la k-moyenne peut générer des résultats divers pour la même valeur de k .

3.3 Extraction des données isolées

Une fois que les k-moyennes atteignent un partitionnement approprié, une mesure spécifique est calculée : la sil-

houette. Elle fait référence à une méthode d'interprétation et de validation de la cohérence dans le regroupement des données et fournit une représentation graphique de la façon dont chaque point a été classé.

Soit $a(i)$ la distance moyenne entre un point i et tous les autres points de données de la même partition ; soit $b(i)$ la plus petite distance moyenne du point i à tout autre groupe dont i ne fait pas partie de l'ensemble. La silhouette du point i est définie comme suit :

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases}$$

la silhouette donne un résultat compris entre -1 et 1 . Un $s(i)$ proche de 1 signifie que les données sont correctement regroupées.

Pour chaque partition, la silhouette est calculée pour chaque point. Si la silhouette de certains points est inférieure à un seuil $0 < silh < 1$, alors ces points sont considérés comme *noise*. Si la silhouette de certains points est supérieure au seuil $silh$, alors ces points sont considérés comme *accessibles*. Si la silhouette fournit une valeur négative, le point est concédé à la partition correspondante. Une nouvelle partition est construite pour chaque point *noise*.

La méthode proposée n'est pas optimale mais linéaire sur le nombre de clusters et le nombre d'itérations, et log-linéaire sur le nombre de points.

4 Opérations sur le jeu de données

Un appareil peut avoir des motifs de consommation qui se produisent dans plusieurs ou une même séquence. Pour limiter la longueur des séquences et améliorer l'efficacité de la prédiction, ces motifs sont regroupés comme un unique symbole, c'est-à-dire qu'une fois le motif détecté ou suggéré, la prédiction le produira dans son entièreté.

Cette section est composée de deux algorithmes : *Disco. Motif. Algo.* est conçu pour découvrir un motif fréquent à l'intérieur de l'ensemble des séquences ; *Ext. Motif. Algo.* étend un motif à l'intérieur des séquences.

4.1 Découverte des motifs

L'exploration de modèle séquentiel est une méthode d'exploration de données importantes qui peut extraire des séquences similaires tout en conservant leur ordre. Cependant, il est essentiel d'identifier les intervalles des éléments des modèles séquentiels extraits par exploration de modèles séquentiels. *Disco. Motif. Algo.* est basé sur l'algorithme de Hirate et Yamana [8].

Ces intervalles peuvent être un intervalle entre deux symboles ou un intervalle de temps. Dans notre jeu de données, les deux approches sont identiques. De plus, le motif doit être continu et contigu. L'exploration de modèle séquentiel est donc définie sur des données contiguës sans intervalle de temps.

En plus du motif, l'algorithme Hirate et Yamara fournit aussi son support. Le support fournit la fréquence de cette

TABLE 1 – Jeu de données après *Clust. Algo.*

ID	Séquence
1	aaabacc
2	bacaaa
3	ccbacaaa

TABLE 2 – Algorithme de Hirate and Yamana.

Motif	Support (>50%)
bac	100%
aaa	100%
ba	100%
...	...
cc	66%
ca	66%

sous-séquence parmi l'ensemble de données mais ne donne pas si le motif est récurrent dans la même séquence. Ce processus est illustré dans les tableaux 1 et 2.

Après avoir trouvé les motifs, seuls ceux dont la longueur et le support sont les plus importants sont pris en compte sans aucun conflit entre eux. Le tableau 3 montre les séquences de l'exemple avant et après traitement. Les motifs sont entre parenthèses.

4.2 Extension des motifs

L'objectif de *Ext. Motif. Algo.* est de découvrir des versions étendues des motifs dans les séquences. Par exemple, les séquences *aaaeaaa* et *aaafaaa* ne diffèrent que par un seul symbole. Si les symboles *e* et *f* ont des valeurs proches, les deux séquences sont considérées comme égales avec un nouveau symbole $g = \frac{e+f}{2}$.

À partir de deux mêmes motifs, l'extension se réalise sur le suffixe et / ou le préfixe. Si la pente du motif généré diffère de α'' pour cent de celui d'origine, alors le processus d'agglomération s'arrête pour le côté concerné.

5 Inférence grammaticale et prédiction

Suivant les étapes précédentes, les cycles de consommation sont désormais composés de symboles et de motifs. L'objectif de cette section est de construire un automate de prédiction à partir de ces séquences. À partir d'un arbre de préfixe de fréquence, une inférence grammaticale construit l'automate stochastique. À ce stade, une marche aléatoire

TABLE 3 – Jeu de données après *Disco. Motif. Algo.*

ID	Séquence
1	(aaa)(bac)c
2	(bac)(aaa)
3	cc(bac)(aaa)

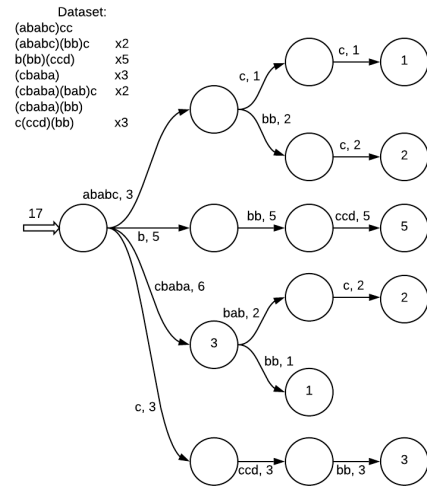


FIGURE 2 – Exemple d'arbre de préfixe.

détermine les schémas de consommation future de l'appareil concerné.

5.1 Arbre de préfixe

Un arbre de préfixe est un arbre dont les transitions représentent les symboles d'une séquence et qui possède les deux propriétés suivantes : pour chaque état, la somme des fréquences (des symboles) entrantes est égale à la somme des fréquences sortantes; le nombre de séquences démarrant à la racine est égal à la somme des arrêts sur les nœuds.

Pour chaque appareil, chaque séquence est entrée dans l'arbre de préfixe comme suit :

1. Pour commencer à analyser une séquence, on se positionne à la racine de l'arbre de préfixe et on incrémente la fréquence d'entrée de 1.
2. Pour chaque symbole d'un mot (ordre des préfixes).
 - (a) Si un arc contenant ce symbole existe déjà, incrémente sa valeur de fréquence de un et continuez d'interpréter le mot sur le nœud enfant correspondant.
 - (b) Sinon, créez un nœud enfant correspondant au symbole avec un arc de valeur de fréquence 1.
 - (c) S'il s'agit du symbole terminal du mot, alors le nœud enfant correspondant devient terminal avec une valeur de terminaison incrémentée de 1.
3. Renvoyer l'arborescence des préfixes de fréquences.

Un exemple est montré sur la figure 2 où chaque mot est suivi de sa fréquence. Les fréquences deviennent des fréquences relatives (probabilité) en divisant chaque fréquence sortante par la somme de toutes les fréquences sortantes.

5.2 Inférence grammaticale

Comme nous avons un arbre de préfixe probabiliste, il est possible de faire une inférence grammaticale sur celui-ci. L'inférence grammaticale a été développée de manière significative par Colin de la Higuera dont les travaux sont présentés dans ces livres [6, 7]. L'inférence grammaticale représente le processus d'apprentissage d'une grammaire formelle à partir d'un ensemble d'observations (dans notre cas, l'arbre de préfixe probabiliste).

Il existe de nombreuses méthodes d'inférence grammaticale, souvent en compétition dans le concours PAutomaC [19]. Parmi les algorithmes, ALERGIA est un algorithme non déterministe permettant la réduction d'un automate déterministe probabiliste par un calcul d'équivalence d'automate probabiliste [4, 18].

ALERGIA a été appliqué à l'extraction d'informations à partir de textes ou de documents structurés et à la modélisation du langage vocal. Lorsque la probabilité d'apparition d'une chaîne suit une distribution bien approximée, ALERGIA a la capacité de fusionner des noeuds lorsque les automates résultants sont compatibles avec la fréquence observée de chaînes.

Soit n_i le nombre de séquences arrivant au noeud i ; f représente la fréquence : $f_i(a)$ est le nombre de séquences arrivant au noeud i et ayant comme prochain symbole a ; $f_i(\cdot)$ est le nombre de séquence se terminant en i . ALERGIA a besoin du calcul préliminaire suivant : $p_i(a) = \frac{f_i(a)}{n_i}$ pour chaque transition sortante.

Deux noeuds i et j sont dit compatibles si pour toutes les transitions (dont les terminaisons) sortantes et récursivement pour l'ensemble de leurs enfants :

$$|p_i(a) - p_j(a)| < \sqrt{0.5 \ln \frac{2}{\epsilon} \left(\frac{1}{\sqrt{n_i}} + \frac{1}{\sqrt{n_j}} \right)}$$

où ϵ est appelé plage d'acceptation.

L'algorithme ALERGIA vérifie toutes les paires de noeuds, si aucune paire n'est compatible, l'algorithme s'arrête. La fusion et la plage d'acceptation sont présentées dans les deux sous-sections suivantes.

5.3 Fusion

Si deux noeuds i et j sont compatibles, ALERGIA traite la fusion et le repli de ceux-ci et de leurs enfants. Ce processus est connu sous le nom d'algorithme RPNI [12].

La fusion de deux noeuds signifie les réduire à un unique, dont la position est la plus petite profondeur des deux fusionnés. Ce noeud est considéré comme terminal si le noeud fusionné était terminal.

Quant aux transitions sortantes, elles sont elles-mêmes fusionnées si elles sont étiquetées avec le même symbole, et dans un tel cas, les deux noeuds pointés sont fusionnés récursivement.

Étant donné que i et j sont fusionnés, dans un premier temps, la transition entre j et ses parents est rompue. Ces transitions se situent actuellement entre i et les parents de j (pour chaque transition).

Ensuite, l'algorithme RPNI absorbe récursivement les enfants de j comme suit :

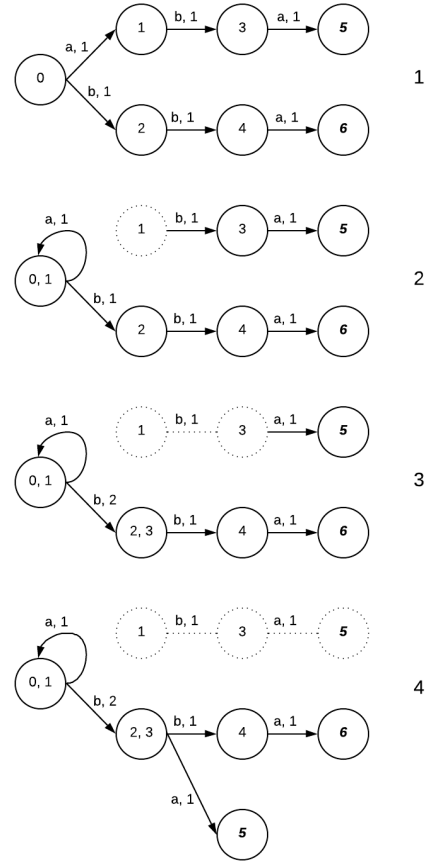


FIGURE 3 – Fusion entre les sommets 0 et 1.

1. Si la transition entre j et un successeur existe entre i et un successeur ; alors la fréquence de ce chemin est incrémentée. Si le successeur de j est un noeud terminal, alors l'état correspondant de i devient ou reste un noeud terminal.
2. Si la transition entre j et un successeur n'existe pas entre i et un successeur, et récursivement pour toutes les transitions entre j et le successeur choisi ; alors le chemin de j au successeur est attaché à i .

Dans la figure 3, nous fusionnons les noeuds 0 et 1. La transition montre le symbole et la fréquence, le noeud terminal est en italique et en gras. À la première étape, la transition entre 1 et ses parents est rompue puis ajoutée entre 0 et le parent de 1, dans ce cas entre 0 et 0. Cela crée une boucle étiquetée a sur le noeud 0 comme indiqué dans la deuxième image.

On considère maintenant le chemin entre 1 et 3, la transition b existe déjà après le noeud 0, mais pas le chemin ba . Ainsi, l'état 3 fusionne avec le noeud 2 (troisième image) mais le noeud 5 reste seul. Ce dernier se réfère à la deuxième règle, le noeud est apposé à l'état 2 comme le montre la quatrième image.

5.4 Influence de la plage d'acceptation

Les scientifiques spécialistes dans l'inférence grammaticale posent la plage d'acceptation à $\epsilon = \frac{1}{N^r}$ avec N le

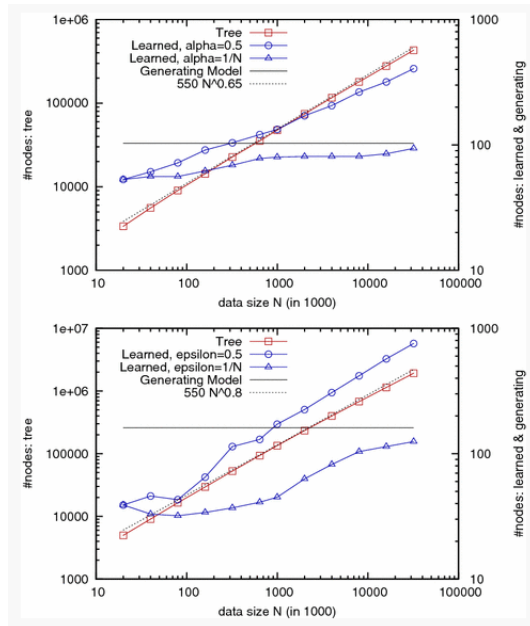


FIGURE 4 – Taille de l’automate finale (en haut $r = 3$, en bas $r = 5$) [14].

nombre de symboles différents dans l’ensemble des séquences, et le facteur de puissance à $r > 2$. Mao et al. [14] ont effectué de nombreuses expériences sur la valeur de r . La figure 4 provient de leur papier, les deux graphes présents indiquent le nombre de sommets de l’automate stochastique en fonction du nombre de sommets initiaux et de la valeur de r .

Afin de comprendre la similitude entre l’automate réduit et l’automate de départ, Mao et al. ont implanté la divergence de Kullback-Leiber [11]. La divergence diminue à r constant et en augmentant le nombre de départ; la divergence augmente quand r augmente (donc si ϵ diminue).

L’automate généré par la présente méthode doit générer les mots de départ mais doit être suffisamment flexible afin de reconnaître des comportements non rencontrés lors de la phase d’apprentissage. Le compromis est trouvé avec $r = 3$.

5.5 Prédiction

Une fois l’automate réduit par l’algorithme ALERGIA, il est possible de l’exploiter pour fournir des prédictions sur la consommation future de l’appareil. Les prédictions sont obtenues en effectuant des marches aléatoires dans l’automate probabiliste sur un certain nombre w d’étapes. Cela est réalisé en considérant l’automate probabiliste comme une chaîne de Markov, le passage à travers un arc produira le symbole associé comme une prédiction. La prédiction est un suffixe de la consommation actuelle.

A noter que la prédiction par marche aléatoire peut se faire en générant des individus. L’autre possibilité, adapté à des chaînes de petite taille, consiste à multiplier le vecteur de population (donc le vecteur de distribution après lecture du préfixe) par la matrice représentant la chaîne de Markov

TABLE 4 – Prédiction des suffixes à partir du mot b .

Longueur	Suffixes et probabilités d’apparition
1	{b, 0.86}; { ϵ , 0.14}
2	{ba, 0.11}; {bb, 0.43}; {b ϵ , 0.32}, { ϵ , 0.14}
3	{bab, 0.07}; {baa, 0.04}; {bb ϵ , 0.43}; {b ϵ , 0.32}; { ϵ , 0.14}

associé à l’automate stochastique.

Étant donné que la fréquence d’un chemin avec des individus et la probabilité d’un suffixe convergent lorsque le nombre d’individus est vaste, ils sont un moyen plus pratique de générer des prévisions. En effet, un arbre de possibilités de longueur croissante utilise plus de mémoire que certaines marches aléatoires. Étant donné que la machine de prédiction peut être intégrée, les marches aléatoires sont préférées.

Expliquons le processus avec un exemple. La figure 5 présente à gauche la machine de prédiction avec les symboles et les fréquences; et à droite la chaîne de Markov correspondante. Le tableau 4 montre une prédiction de l’augmentation de la longueur du mot b avec la probabilité correspondante du suffixe (où ϵ signifie la fin du mot).

6 Expériences

6.1 Résultats

Les tests ont été effectués avec un ensemble de données de plus de 700.000 lignes référençant la quantité de kW consommée avec un timer d’une seconde, soit plus d’une semaine de données. Le dispositif étudié est un réfrigérateur industriel fréquemment utilisé et pouvant contenir des produits différents d’une journée à l’autre. Sa consommation dépend des éléments à l’intérieur, de l’heure d’ouverture de la porte et de sa variable environnementale. La figure 6 présente un échantillon de l’ensemble de données.

La prédiction a été générée plusieurs fois et toujours après un départ de consommation de l’appareil. Toutes les prédictions fournissent une prévision précise. En prenant un intervalle de 95% des prédictions faites à partir d’un instant donné (nous écartons les prédictions trop éloignées de la masse), au moins 92% (97% en moyenne) de toutes les prévisions sont similaires à la consommation réelle avec un écart-type d’au plus 5% (3% en moyenne). Cet écart-type s’explique par les différentes approximations faites dans la méthode proposée; l’écart type dépend de la valeur des seuils.

6.2 Discussions

La méthode présentée dans ce papier obtient de bons résultats sur des appareils ayant une latence dans sa consommation. C’est-à-dire que suite à un stimulus, la consommation de l’appareil est perturbé après et pendant un certain temps qui est bien plus grande que la durée du stimulus. En effet, l’automate de prédiction est une méthode assez simple

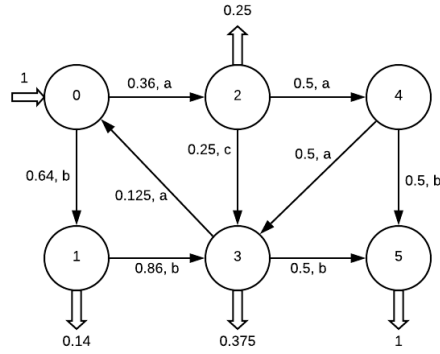
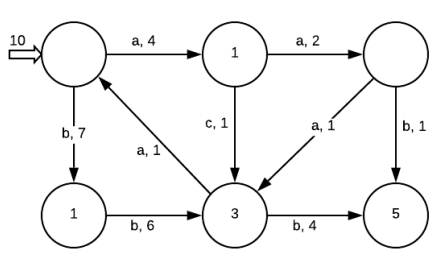


FIGURE 5 – De l’automate à fréquence à un automate stochastique.

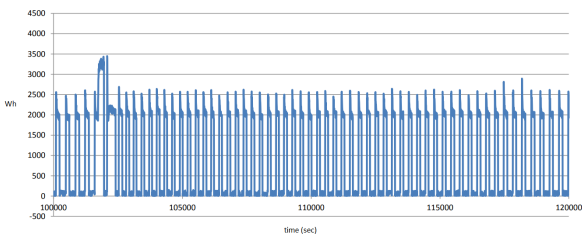


FIGURE 6 – Données utilisées.

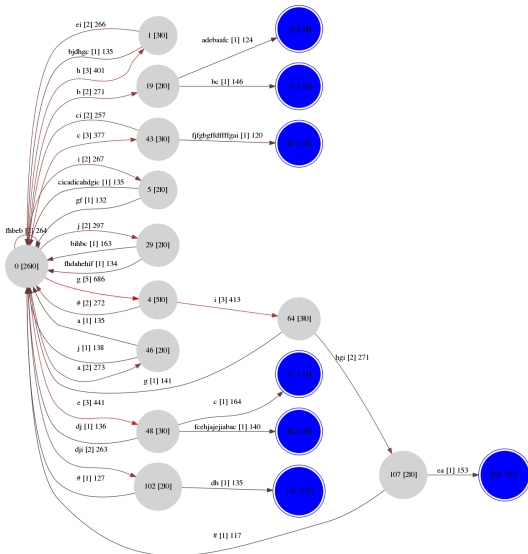


FIGURE 7 – Automate stochastique obtenu par la méthode présentée.

représentant des cycles de consommations. Nous avons vu précédemment que le processus de la méthode repère les schémas récurrents puis réduit l’automate par inférence. Il est donc logique que l’automate soit capable de reconnaître des schémas à des types de stimuli mais ait plus de difficulté à prédire la consommation d’un appareil plus erratique.

Si l’automate présente des résultats non satisfaisants, il est possible de créer un nouvel automate stochastique avec des nouvelles valeurs.

7 Conclusion

La prédiction de la consommation est un problème fondamental du réseau électrique intelligent. La plupart des méthodes de la littérature font appel à des méthodes coûteuses, consommatrices mais ayant des résultats similaires à la réalité. Cependant, notre approche soulève et répond à une problématique sous-jacente à la prédiction : peut-on prédire la consommation future avec peu de mémoire et une puissance de calcul minimale ? si possible de manière embarquée dans l’appareil ou chez le consommateur.

La méthode proposée est adaptée à des appareils ayant des schémas distincts et répondant à des stimuli avec une grande amplitude dans le temps comparé à la durée du stimulus. Le principal défaut de la méthode est la mise à jour de son automate stochastique en repartant de zéro. Les travaux futurs ont pour objectif de changer l’automate stochastique en chaîne de Markov à état caché. En effet ces dernières sont capables d’apprendre les nouveaux cycles de consommation au fil de l’eau si la prédiction s’avère trop médiocre.

De plus, la première partie de l’algorithme sur la discrétisation des données doit être retravaillée et comparée à des méthodes comme [3], SAX [13] et Persist [15]

Références

- [1] AS Ahmad, MY Hassan, MP Abdullah, HA Rahman, F Hussin, H Abdullah, and R Saidur. A review on applications of ann and svm for building electrical energy consumption forecasting. *Renewable and Sustainable Energy Reviews*, 33 :102–109, 2014.
- [2] Kadir Amasyali and Nora M El-Gohary. A review of data-driven building energy consumption prediction studies. *Renewable and Sustainable Energy Reviews*, 81 :1192–1205, 2018.
- [3] Marc Boullé. Modl : A bayes optimal discretization method for continuous attributes. *Machine learning*, 65(1) :131–165, 2006.
- [4] Rafael C Carrasco and José Oncina. Learning stochastic regular grammars by means of a state merging method. In *International Colloquium on Grammatical Inference*, pages 139–152. Springer, 1994.

- [5] Mohammad Azhar Mat Daut, Mohammad Yusri Hassan, Hayati Abdullah, Hasimah Abdul Rahman, Md Pauzi Abdullah, and Faridah Hussin. Building electrical energy consumption forecasting analysis using conventional and artificial intelligence methods : A review. *Renewable and Sustainable Energy Reviews*, 70 :1108–1118, 2017.
- [6] Colin De la Higuera. *Grammatical inference : learning automata and grammars*. Cambridge University Press, 2010.
- [7] Rémi Eyraud, Colin De La Higuera, Makoto Kanazawa, and Ryo Yoshinaka. Introduction to the grammatical inference special issue of *fundamenta informatica*, 2016.
- [8] Yu Hirate and Hayato Yamana. Generalized sequential pattern mining with item intervals. *JCP*, 1(3) :51–60, 2006.
- [9] Tao Hong and Shu Fan. Probabilistic electric load forecasting : A tutorial review. *International Journal of Forecasting*, 32(3) :914–938, 2016.
- [10] Han Jaiwei and Micheline Kamber. Data mining : concepts and techniques. ed : *Morgan Kaufmann San Francisco*, 2006.
- [11] Solomon Kullback. *Information theory and statistics*. Courier Corporation, 1997.
- [12] Kevin J Lang. Random dfa’s can be approximately learned from sparse uniform examples. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 45–52, 1992.
- [13] Jessica Lin, Eamonn Keogh, Li Wei, and Stefano Lonardi. Experiencing sax : a novel symbolic representation of time series. *Data Mining and knowledge discovery*, 15(2) :107–144, 2007.
- [14] Hua Mao, Yingke Chen, Manfred Jaeger, Thomas D Nielsen, Kim G Larsen, and Brian Nielsen. Learning deterministic probabilistic automata from a model checking perspective. *Machine Learning*, 105(2) :255–299, 2016.
- [15] Fabian Mörchen and Alfred Ultsch. Finding persisting states for knowledge discovery in time series. In *From Data and Information Analysis to Knowledge Engineering*, pages 278–285. Springer, 2006.
- [16] Muhammad Qamar Raza and Abbas Khosravi. A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings. *Renewable and Sustainable Energy Reviews*, 50 :1352–1372, 2015.
- [17] Jason Runge and Radu Zmeureanu. Forecasting energy use in buildings using artificial neural networks : a review. *Energies*, 12(17) :3254, 2019.
- [18] Franck Thollard, Pierre Dupont, Colin de la Higuera, et al. Probabilistic dfa inference using kullback-leibler divergence and minimality. In *ICML*, pages 975–982, 2000.
- [19] Sicco Verwer, Rémi Eyraud, and Colin De La Higuera. Pautomac : a probabilistic automata and hidden markov models learning competition. *Machine learning*, 96(1-2) :129–154, 2014.
- [20] Zeyu Wang and Ravi S Srinivasan. A review of artificial intelligence based building energy use prediction : Contrasting the capabilities of single and ensemble prediction models. *Renewable and Sustainable Energy Reviews*, 75 :796–808, 2017.
- [21] Yixuan Wei, Xingxing Zhang, Yong Shi, Liang Xia, Song Pan, Jinshun Wu, Mengjie Han, and Xiaoyun Zhao. A review of data-driven approaches for prediction and classification of building energy consumption. *Renewable and Sustainable Energy Reviews*, 82 :1027–1047, 2018.
- [22] Hai-xiang Zhao and Frédéric Magoulès. A review on the prediction of building energy consumption. *Renewable and Sustainable Energy Reviews*, 16(6) :3586–3592, 2012.
- [23] Kasım Zor, Oğuzhan Timur, and Ahmet Teke. A state-of-the-art review of artificial intelligence techniques for short-term electric load forecasting. In *2017 6th International Youth Conference on Energy (IYCE)*, pages 1–7. IEEE, 2017.