



Improving performance of non-intrusive load monitoring with low-cost sensor networks

Xuan-Chien Le

► To cite this version:

Xuan-Chien Le. Improving performance of non-intrusive load monitoring with low-cost sensor networks. Signal and Image processing. Université de Rennes, 2017. English. NNT : 2017REN1S019 . tel-01622355

HAL Id: tel-01622355

<https://theses.hal.science/tel-01622355>

Submitted on 24 Oct 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Bretagne Loire

pour le grade de
DOCTEUR DE L'UNIVERSITÉ DE RENNES 1
Mention : Traitement du Signal et Télécommunications
École doctorale MATISSE

présentée par
Xuan-Chien LE

préparée à l'unité de recherche UMR6074 IRISA
Institut de recherche en informatique et systèmes aléatoires - CAIRN
École Nationale Supérieure des Sciences Appliquées et de Technologie

**Improving
Performance of
Non-Intrusive
Load Monitoring
with Low-Cost
Sensor Networks**

**Thèse soutenue à Lannion
le 12 avril 2017**

devant le jury composé de :

Jean-François DIOURIS

Professeur à l'Université de Nantes / rapporteur

Laurent CLAVIER

Professeur à l'Institut Mines Telecom / rapporteur

Seddik BACHA

Professeur à l'Université de Grenoble / examinateur

David MENGA

Ingénieur Chercheur à EDF Lab / examinateur

Benoît LEPRETTRE

Ingénieur Chercheur à Schneider Electric / examinateur

Charlotte LANGLAIS

Maître de Conférences à Telecom Bretagne / examinateur

Olivier SENTIEYS

Professeur à l'Université de Rennes 1 / directeur de thèse

Baptiste VRIGNEAU

Maître de Conférences à l'Université de Rennes 1 / co-directeur de thèse

"Ever tried. Ever failed. No matter. Try Again. Fail again. Fail better."

— Samuel Beckett

UNIVERSITE DE RENNES 1

Résumé étendu

Ecole Doctorale MATISSE

Université de Rennes 1

Docteur en Philosophie

**Amélioration des Performances de Supervision de Charges Non Intrusive à
l'Aide de Capteurs Sans Fil Faible Coût**

par [Xuan-Chien LE](#)

Dans un futur proche, il sera intéressant de surveiller la consommation électrique des différents équipements d'une maison ou d'un bâtiment dans un but d'optimisation et de réduction de l'énergie, tout en assurant une qualité de confort à l'utilisateur. Idéalement, l'intervention humaine doit être limitée au maximum avec un système qui s'adapte automatiquement à son environnement.

La réalisation d'un tel système demande de récupérer et de centraliser régulièrement des informations sur l'environnement comme la présence de personnes, la température, le taux d'humidité, la luminosité, etc.

Il est facile de récolter et d'observer ces informations grâce à des capteurs. Cependant, la surveillance de la charge électrique est plus compliquée à réaliser. Des solutions intrusives consistent à superviser individuellement les différents appareils avec un wattmètre, nécessitant une coupure de l'alimentation puis le déploiement à tous les appareils et enfin un entretien régulier ensuite. La solution intrusive n'est pas envisageable pour des raisons pratiques.

Au lieu d'intercaler un wattmètre, le coût de déploiement peut être réduit en utilisant des capteurs indirects pour observer l'environnement et estimer la consommation électrique. Cependant, cette solution demande encore un nombre important de capteurs engageant des interventions techniques régulières.

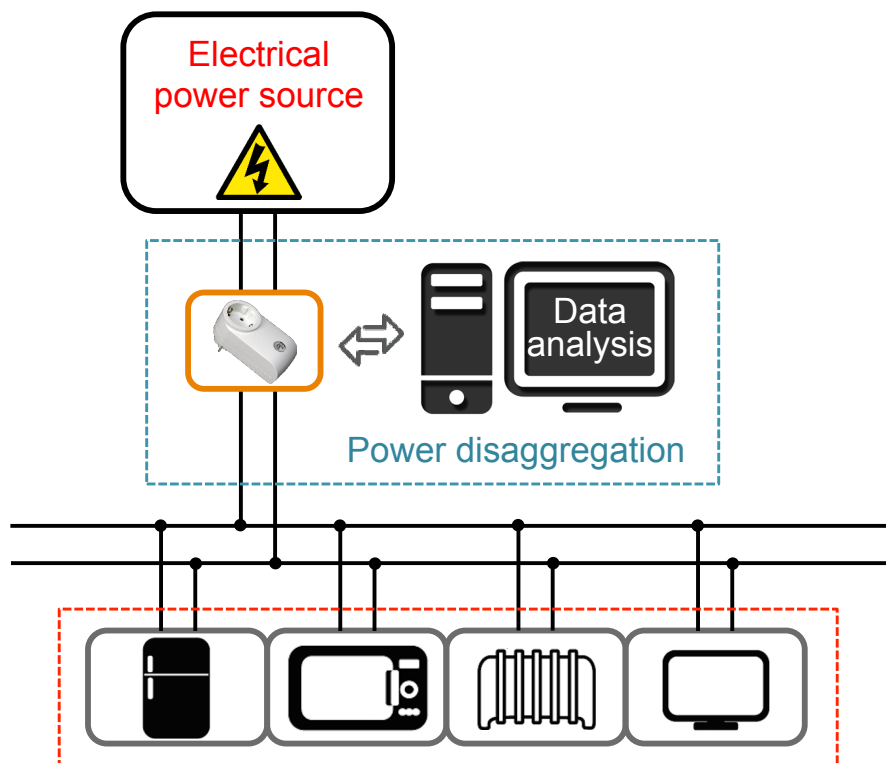


FIGURE 1: Système de monitoring non intrusif d'appareil ou NILM.

C'est pourquoi G. Hart [1] a introduit le concept de monitoring non intrusif d'appareils, NIALM (Non-Intrusive Appliance Load Monitoring) ou NILM (Non Intrusive Load Monitoring), représenté sur la Figure 1, où le fonctionnement d'un appareil est estimé à partir de la puissance totale issue de l'aggrégation des consommations d'un panel d'appareils. Les résultats issus d'un relevé de mesures unique sont encourageants.

Les algorithmes utilisés dépendent de la fréquence d'échantillonnage des informations données par le wattmètre. Concrètement, une solution matérielle à faible fréquence pourra se baser sur des valeurs moyennes ou des échelons de variations alors qu'une fréquence élevée exploitera les variations transitoires ou une analyse fréquentielle issue de la transformée de Fourier.

Néanmoins, la principale raison d'erreur de cette approche provient de l'ambiguïté créée par les appareils possédant des caractéristiques électriques très proches.

Pour relever ce défi, les travaux de cette thèse proposent d'associer au NILM une information extérieure pour améliorer les performances des algorithmes.

La première proposition allie une solution basée sur la transition entre l'état précédent et l'état actuel de chaque appareil. La meilleure combinaison d'états des appareils est alors sélectionnée en résolvant la minimisation de la norme $l1$ avec une recherche exhaustive. L'information extérieure exploitée est la probabilité de l'état de l'appareil fournie par un réseau de capteurs sans fil. Positionné entre le NILM non intrusif et les solutions intrusives, ce système est dénommé SmartSense et peut ne superviser qu'une partie des appareils à l'aide du réseau de capteurs.

Approche de la norme $l1$: une approche naïve du NILM pour déterminer les états des appareils est d'appliquer une résolution exhaustive (*brute force*) pour minimiser le problème de norme $l1$ défini comme

$$\begin{aligned} \min_{\mathbf{s}} \quad & \| x(t) - \sum_{i=1}^N \sum_{j=1}^{m_i} s_{ij}(t) \times w_{ij} \| \\ \text{sujet à} \quad & s_{ij}(t) \in \{0, 1\}, i = 1, \dots, N, j = 1, \dots, m_i \\ & \sum_{j=1}^{m_i} s_{ij}(t) \in \{0, 1\}, i = 1, \dots, N, \end{aligned}$$

où $x(t)$ est la puissance agrégée à l'instant t , N est le nombre d'appareils, m_i est le nombre d'états de l'appareil i , w_{ij} et $s_{ij}(t)$ représentent la puissance consommée et la variable booléenne de l'état j de l'appareil i . Dans la résolution par force brute, toutes les combinaisons possibles des états des appareils sont appliquées successivement au problème de minimisation en calculant les erreurs entre la puissance agrégée et la

puissance totale correspondante à la combinaison considérée. La solution correspond à la combinaison donnant la plus petite erreur.

En conditions réelles, la consommation d'un appareil n'est pas stable et considérée comme bruitée. Il faut alors éviter que la bonne combinaison ne soit éliminée et pour cela, nous avons proposé de retenir un lot de combinaisons dont l'erreur est proche du minimum. À ce moment, le choix final utilise l'information extérieure et exploite les relations entre l'état présent et l'état précédent selon deux critères proposés. Le premier est la distance de Hamming puisque nous avons des vecteurs binaires dans notre modèle. La combinaison donnant la plus petite distance de Hamming avec le vecteur état précédent sera solution de l'algorithme.

Le second critère propose de considérer la probabilité de transition entre l'état précédent et les combinaisons candidates. La solution sera la combinaison avec la probabilité la plus élevée. La probabilité est estimée durant une période d'apprentissage.

Pour évaluer les performances de ces nouvelles solutions, nous avons une base de données de consommation issue d'un déploiement interne au laboratoire. Les simulations à l'aide de Matlab montrent que la précision, considérée comme la fiabilité de détection, est améliorée de 7% avec la distance de Hamming et de 19% avec la probabilité de transition. Les améliorations en considérant le rappel, représentatif de la sensibilité des algorithmes à l'événement, sont de 2% et 18%.

Le système SmartSense : le principe repose sur l'estimation de la probabilité de fonctionnement des appareils comme information extérieure. Un réseau de capteurs sans fil faible coût et faible consommation permet de remonter l'information, comme représenté sur la Figure 2. Cependant, l'intrusion doit être limitée, contrairement aux systèmes intrusifs décrits précédemment, en créant une diversité de types de capteurs non dédiée à une cible et la possibilité de restreindre la surveillance à seulement certains appareils.

Les équipements non supervisés ont alors des états considérés équiprobables. Cependant, l'information partielle concernant les appareils supervisés permet d'améliorer les performances globales des algorithmes de désaggrégation. Pour en estimer le gain, deux approches sont appliquées sur trois bases de données différentes (notre base de données et deux autres mises en partage).

La première approche considère les algorithmes basés sur les détections de variations (Edge Detection ED) [1] et la déformation temporelle dynamique (Dynamic Time Warping DTW) [2]. Un détecteur de variations est basé sur une reconnaissance d'une paire de

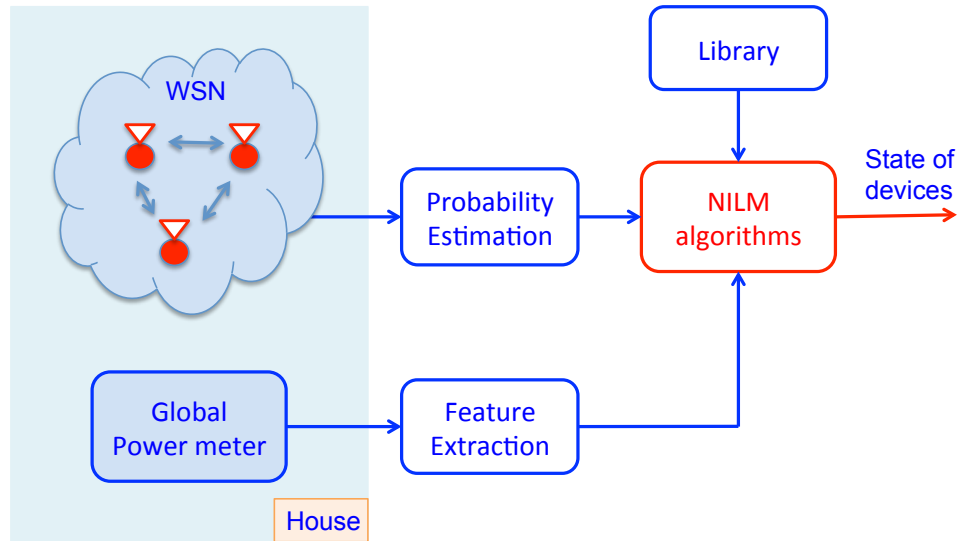


FIGURE 2: Le système SmartSense associe un NILM classique utilisant la puissance globale avec un réseau de capteurs sans fil permettant l’apport d’une information extérieure (ici la probabilité de fonctionnement d’équipements).

fronts montant et descendant correspondant à l’activation d’une charge dans la consommation totale. Une base de données est alors utilisée pour la reconnaissance de l’appareil. Pour ce faire, le motif formé par les deux fronts est comparé à l’aide d’une distance aux éléments de la bibliothèque obtenue à partir de la période d’apprentissage. L’appareil sélectionné est celui correspondant à la plus petite distance.

La difficulté rencontrée par cette solution est liée au fait que beaucoup d’appareils possédant les mêmes fronts sont présents dans les maisons et les immeubles, ayant comme conséquence directe de détériorer les performances de désagrégation. Ainsi, SmartSense exploite l’information extérieure de probabilité de mise en route ou d’extinction d’un appareil au même instant qu’une détection de fronts.

La seconde possibilité basée sur le DTW considère toutes les valeurs comprises dans la fenêtre temporelle formée par les fronts montant et descendant. Comme les longueurs de motifs dépendent de la durée d’activation de l’appareil, la distance cumulée est utilisée lors de la recherche et la comparaison avec les éléments de la bibliothèque. Lorsque SmartSense utilise l’algorithme de DTW, la probabilité de mise en route de chaque appareil est intégrée dans le calcul de la distance cumulée comme un paramètre de régularisation.

L’autre approche appliquée à SmartSense est la formulation du problème de minimisation de la norme l_1 du NILM comme un problème de sac à dos, puis d’utiliser soit une solution heuristique (Compositional Pareto-algebraic Heuristic CPH) [3], soit un algorithme

de programme dynamique (Dynamic Program DP). Dans cette modélisation, chaque appareil est considéré comme un élément d'un groupe avec son propre poids (puissance demandée) et le profit calculé à partir du logarithme de la probabilité et de la demande de puissance. Il faut alors sélectionner les éléments de chaque groupe afin de maximiser le profit sous la contrainte de la capacité du sac à doc, c'est-à-dire la puissance agrégée. Comme précédemment, la probabilité des appareils supervisés est estimée grâce à un réseau de capteurs sans fil et les appareils restants ont des états équiprobables.

L'algorithme CPH a une forme récursive. À chaque itération, toutes les combinaisons possibles de deux appareils sont appairées puis les poids et profits sont calculés. Une partie des solutions est éliminée si leurs poids dépassent la capacité du sac à dos ou si elles sont moins intéressantes, c'est-à-dire qu'elles consomment plus mais avec une valeur de profit inférieure. Quand tous les appareils ont été pris en compte et les solutions possibles listées, la combinaison offrant le plus grand profit est choisie comme solution finale. En outre, la procédure est récursive et est basée sur la relation entre le meilleur profit d'un sous ensemble d'appareils en sélectionnant ou rejetant le dernier élément considéré depuis ceux restants à intégrer. Pour ce faire, une table de profit est construite afin d'établir cette relation de récurrence entre chaque ligne qui représente un appareil et les colonnes qui représentent le remplissage du sac avec des valeurs croissantes de 0 à la capacité maximale. La valeur de chaque cellule est calculée grâce à la relation de récurrence et le meilleur profit de la dernière cellule. La liste finale des appareils est alors construite en remontant la table des profits.

Les simulations obtenues avec Matlab et plusieurs bases de données montrent que les performances des algorithmes sont significativement améliorées lorsqu'ils utilisent la probabilité des états d'un appareil. Concrètement, comme représenté sur la Figure 3, dans le cas de la *maison 1* des relevés de REDD [4], la surveillance d'un réfrigérateur et d'un lave-vaisselle entraîne une augmentation de la précision de 20% et du rappel de 15%. À titre de comparaisons, l'ED améliore de 6% et 4,5% et le DTW de 4% et 2,75%. Ces dernières performances sont limitées et dépendent directement des possibilités du détecteur de front. Pour SmartSense, plus le nombre d'appareils supervisés augmente et meilleures sont les performances. Cependant, certains appareils n'auront qu'une aide limitée et il faut cibler ceux qui entraînent des ambiguïtés et qui ont une fréquence élevée de fonctionnement.

Dans cette thèse, nous montrons le potentiel de la combinaison d'une information extérieure avec les algorithmes de NILM proposés dans la littérature. Malgré quelques limitations comme un apprentissage et une complexité encore trop élevée, SmartSense

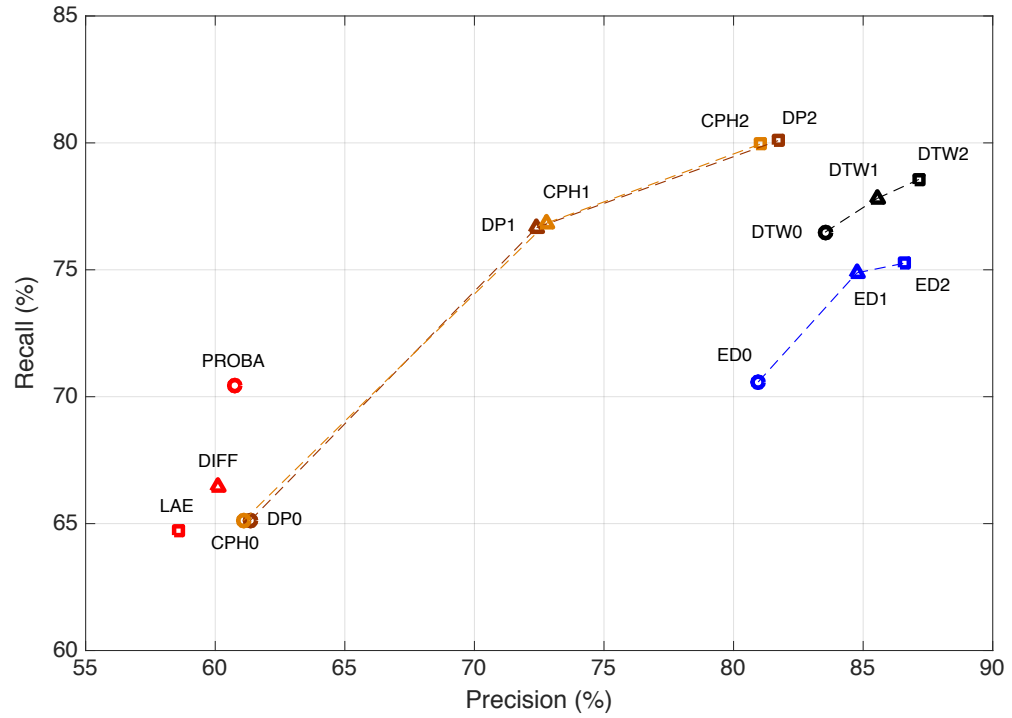


FIGURE 3: Performances des algorithmes dans le cas de la *maison 1* des relevés de REDD [4]. L'indice suivant les algorithmes indique le nombre de dispositifs surveillés : 0 correspond à un NILM classique, 1 et 2 correspondent à la surveillance des appareils offrant les meilleurs résultats.

est développé et déployé en ce moment et des travaux futurs pourront intégrer des approches plus efficaces comme les modèles de Markov cachés ou l'apprentissage profond de réseaux de neurones.

Acknowledgements

More than three years working in this thesis is the most memorial period in my life. It is an honor for me to work and study in a dynamic laboratory, CAIRN in Lannion.

Firstly, I grateful thank my supervisor Oliver Sentieys, the head of CAIRN, and co-supervisor Baptiste Vrigneau for their consistent supervision and encouragement throughout my study and research, their patient guidance on direction of my work. They gave me breakthrough ideas to improve my works. All of their feedback and suggestions at various stages have been significantly improved my knowledge, scientific minds, and skills of writing and presentation. Besides, I would like to thank Arnaud Carer for his supports in sensor network deployment at the first step of my work.

Secondly, I would like to thank all of my colleagues at ENSSAT and all of my friends in Lannion, especially the Vietnamese community, for their sharing and help. This helps me alleviate homesickness and focus on my work.

My warmest thanks also go to my parents and my younger sister for their never ending support. They always assist me in implementing my desires.

Finally, I would like to say something special to my wife. She is a powerful emotional support that I can share my burdens and sadness. Thank you for her appearance in my life, her understanding, her help and her love.

Contents

| | |
|--|-----------|
| Résumé étendu | II |
| Acknowledgements | i |
| Abbreviations | iv |
| List of Figures | vi |
| List of Tables | ix |
| 1 Introduction | 1 |
| 1.1 Smart Grid and Smart Building Automation | 1 |
| 1.2 Load Monitoring | 4 |
| 1.2.1 Intrusive Load Monitoring | 4 |
| 1.2.2 Non-Intrusive Load Monitoring | 4 |
| 1.2.3 Hybrid Load Monitoring | 7 |
| 1.3 Thesis Outline and Contributions | 7 |
| 2 State of the Art | 10 |
| 2.1 Intrusive Load Monitoring | 10 |
| 2.2 Non-Intrusive Load Monitoring | 14 |
| 2.2.1 Low frequency approaches | 15 |
| 2.2.2 High frequency approaches | 24 |
| 2.3 Hybrid Load Monitoring | 26 |
| 2.4 Conclusions | 27 |
| 3 Algorithms for NILM based on l_1-norm Minimization | 35 |
| 3.1 Least Absolute Error based Algorithm | 36 |
| 3.2 State Difference based Algorithm | 37 |
| 3.3 State Transition Probability based Algorithm | 38 |
| 3.4 Experimental Results | 40 |
| 3.4.1 Athemium data collection | 40 |
| 3.4.2 Results and Evaluation | 42 |
| 3.5 Conclusions | 47 |
| 4 SmartSense: Sensor-Aided Non-Intrusive Load Monitoring | 49 |
| 4.1 SmartSense Principle and Model | 49 |
| 4.2 Approach 1: Formulation as a Knapsack problem | 51 |

| | | |
|----------|--|-----------|
| 4.2.1 | Problem formulation | 52 |
| 4.2.2 | Algorithms to solve the SmartSense problem | 55 |
| 4.2.2.1 | Compositional Pareto-Algebraic Heuristic (CPH) | 55 |
| 4.2.2.2 | Dynamic Programming (DP) | 59 |
| 4.3 | Approach 2: Edge Detector Algorithms | 64 |
| 4.3.1 | Edge Detection (ED) | 65 |
| 4.3.2 | Dynamic Time Warping (DTW) | 66 |
| 5 | Experimental Results on SmartSense | 71 |
| 5.1 | Dataset and Metrics | 71 |
| 5.2 | Results and Evaluation of Probability Information in SmartSense | 74 |
| 5.2.1 | Performance without probability information | 74 |
| 5.2.2 | Impact of type and number of monitored devices in the Knapsack approach | 77 |
| 5.2.3 | Impact of WSN detection on the performance of the Knapsack approach | 81 |
| 5.2.4 | Impact of monitored devices on the performance of the edge detector approach | 82 |
| 5.2.5 | Performance comparison between the two approaches | 86 |
| 5.2.6 | Complexity analysis | 89 |
| 5.3 | Conclusions | 93 |
| 6 | Conclusions and Perspectives | 95 |
| 6.1 | Conclusions | 95 |
| 6.2 | Perspectives | 98 |

Abbreviations

| | |
|--------|---|
| ANN | Artificial Neural Network |
| BiRNN | Bidirectional Recurrent Neural Network layer |
| BLUED | Building-Level fUlly-labeled dataset for Electricity Disaggregation |
| BPN | Back-Propagation neural Network |
| CFHSMM | Conditional Factorial Hidden Semi-Markov Model |
| CONV | CONVolutional layer |
| DAE | Denoising Autoencoder |
| DHW | Domestic Hot Water |
| DT | Decision Tree |
| DTW | Dynamic Time Warping |
| ED | Edge Detection |
| EMI | Electromagnetic Interference |
| EMS | Energy Management System |
| FC | Fully Connected layer |
| FFT | Fast Fourier Transform |
| HLM | Hybrid Load Monitoring |
| ILM | Intrusive Load Monitoring |
| LAE | Least Absolute Error |
| LSTM | Long Short-Term Memory |
| HMM | Hidden Markov Model |
| HVAC | Heating, Ventilation and Air Conditioning |
| MLE | Maximum Likelihood Estimation |
| NIALM | Non-Intrusive Appliance Load Monitoring |
| NILM | Non-Intrusive Load Monitoring |
| NIOM | Non-Intrusive Occupancy Monitoring |
| PIR | Passive Infrared Sensor |
| PLC | Power-Line Communication |
| RECAP | Real-time Electrical Appliance Recognition |
| REDD | Reference Energy Disaggregation Dataset |
| RMS | Root Mean Square |

| | |
|------------|--|
| RNN | Recurrent Neural Network |
| RS | Reed Switch |
| SBA | Smart Building Automation |
| SG | Smart Grid |
| SmartSense | Sensor-aided Non-Intrusive Load Monitoring |
| SOM | Self-Organizing Map |
| UK-DALE | UK Domestic Appliance-Level Electricity |
| WSN | Wireless Sensor Networks |

List of Figures

| | | |
|------|--|------|
| 1 | Système de monitoring non intrusif d'appareil ou NILM. | III |
| 2 | Le système SmartSense associe un NILM classique utilisant la puissance globale avec un réseau de capteurs sans fil permettant l'apport d'une information extérieure (ici la probabilité de fonctionnement d'équipements). | VI |
| 3 | Performances des algorithmes dans le cas de la <i>maison 1</i> des relevés de REDD [4]. L'indice suivant les algorithmes indique le nombre de dispositifs surveillés : 0 correspond à un NILM classique, 1 et 2 correspondent à la surveillance des appareils offrant les meilleurs résultats. | VIII |
| 1.1 | Principle components of a SBA system. | 2 |
| 1.2 | Variation on the power signal when the occupant turns on some devices [5]. | 3 |
| 1.3 | Direct and indirect power measurement [6]. | 5 |
| 1.4 | Different categories of load based on energy patterns [7]. | 6 |
| 2.1 | Electromagnetic field waveform. Top: Overhead fan is powered up at point (a), switched on at point (b) and switched off at point (c). Bottom: Variations in magnetic field near by a desktop computer [8]. | 11 |
| 2.2 | The ViridiScope system considers the total power consumption, magnetic field signal and internal power state information to estimate the power consumption of each device by solving an l_1 -norm minimization problem [6]. | 13 |
| 2.3 | Power meter topology. The left topology uses only one power meter (dark node) at the main power supply while topology 1 and 2 use two additional power meters at the electrical circuit branches [9]. | 14 |
| 2.4 | Step-changes on the aggregate power signal [1]. | 16 |
| 2.5 | Load distribution in P-Q plane [10]. | 16 |
| 2.6 | ANN to identify the water heater (group H) and washing machine/dish washer (group W) [11]. | 20 |
| 2.7 | Examples of convolutional neural network, bidirectional RNN and fully connected neural network. | 28 |
| 2.8 | Outputs produced by three deep neural networks [12]. | 29 |
| 2.9 | Difference HMM [13]. Shaded nodes represent the observations and unshaded nodes represent the hidden variables. | 29 |
| 2.10 | On-duration histogram of some devices [14]. | 30 |
| 2.11 | Graphical representation of CFHSM with the additional conditions $c^{(1)}$, $c^{(2)}$ [14]. | 30 |
| 2.12 | A computer and an incandescent light bulb cannot be discriminated by the real and reactive power but can be distinguished by analyzing the third harmonic [15]. | 31 |
| 2.13 | v -shapes on the transient of an induction motor [16]. | 31 |

| | | |
|------|---|----|
| 2.14 | Harmonic signatures of typical devices [17]. | 32 |
| 2.15 | (a) Current and voltage waveforms of a television. (b) Corresponding V-I trajectory [18]. | 32 |
| 2.16 | New noise feature extracted from the background noise [19]. | 33 |
| 2.17 | Time-varying EMI of a laptop at idle, medium load and high load [20]. | 33 |
| 2.18 | Power consumption with and without the occupants [21]. | 33 |
| 2.19 | Aggregate power consumption and the variation of audio signal in a living room [22]. | 34 |
| 3.1 | State transition probability of some devices in Athemium dataset obtained from a training period by attaching each of them to a power meter. | 39 |
| 3.2 | Smart meters network to retrieve the Athemium data. | 40 |
| 3.3 | Athemium web interface. | 41 |
| 3.4 | Athemium power measurement: a global power meter measures the aggregate power consumption and some sub-meters provide training data and ground truth data of each device. | 42 |
| 3.5 | Power measurement of the devices in Athemium dataset during one day. | 43 |
| 3.6 | Performance of the proposed algorithms including LAE, state difference based (DIFF) and transition probability based (PROBA) compared with edge detection algorithm (ED) applied to Athemium data. | 44 |
| 3.7 | Effect of threshold γ on the performance of state transition probability based algorithm simulated with Athemium dataset. | 46 |
| 3.8 | Performance of the proposed algorithms including LAE, state difference based (DIFF) and transition probability based (PROBA) compared with the edge detection (ED) algorithm applied to REDD <i>House 1</i> | 47 |
| 4.1 | SmartSense system model: an NILM system combining with a sensor network providing the operating probability of some specific devices. | 50 |
| 4.2 | Running example of CPH. (a) Three devices with corresponding power demand and probability of each state as well as the aggregate power consumption. (b) Each device is represented as a set of tuples, each tuple corresponds to a state. At each iteration, a partial solution can be discarded if its first element (power consumption) exceeds the bound R (rule 1), or its third element is greater than another solution while consuming larger or same power (rule 2). After all sets are combined, the tuple giving the smallest value of the third element is selected as the final solution. | 56 |
| 4.3 | CPH algorithm flowchart. All sets of tuples are saved in cell D . After the last iteration, the solution is extracted by selecting a tuple in C with the least value of the third dimension. | 57 |
| 4.4 | DP algorithm flowchart. When all entries are filled, the corresponding state vector is obtained by backtracking through the profit tables from the last cell (optimal value). | 62 |
| 4.5 | Load disaggregation based on the ED algorithm. | 67 |
| 4.6 | DTW pattern. | 68 |
| 4.7 | Combination of DTW and ED algorithms. A threshold β is used to reject a pattern if it does not match with any one in DTWlib | 70 |

| | | |
|------|--|----|
| 5.1 | Effect of the type of monitored devices on the performance improvement of the CPH and DP algorithms. The values of pr_0 , rc_0 , Fm_0 correspond to the case without any monitored device, given in Table 5.6. | 77 |
| 5.2 | Effect of the number of monitored devices on the performance improvement of the CPH and DP algorithms. The values of pr_0 , rc_0 , Fm_0 correspond to the case without any monitored device, given in Table 5.6. . . . | 80 |
| 5.3 | Performance improvement of the CPH algorithm when modifying WSN precision. | 81 |
| 5.4 | Dependence of the abnormally changing point on the number of states. . | 81 |
| 5.5 | Effect of the type of monitored devices on the performance improvement of the DTW and ED algorithms. The values of pr_0 , rc_0 , Fm_0 correspond to the case without any monitored device, given in Table 5.6. | 83 |
| 5.6 | Effect of the number of monitored devices on the performance improvement of the DTW and ED algorithms. The values of pr_0 , rc_0 , Fm_0 correspond to the case without any monitored device, given in Table 5.6. . . . | 84 |
| 5.7 | Performance improvement of the ED algorithm versus the WSN precision. | 85 |
| 5.8 | Performance comparison of the proposed algorithms with two monitored devices. | 86 |
| 5.9 | Effect of the number of Pareto points on the performance and complexity of the CPH algorithm. | 91 |
| 5.10 | Effect of the scale factor on the performance and complexity of the DP algorithm. | 92 |
| 6.1 | Performance of all proposed algorithms with REDD dataset <i>House 1</i> . The index following the algorithms denotes the number of monitored devices. | 97 |

List of Tables

| | | |
|------|---|----|
| 2.1 | Performance comparison between DTW and HMM approaches [2]. | 18 |
| 3.1 | Power demand of devices in Athemium dataset. | 42 |
| 3.2 | Detailed performance of the three proposed algorithms including LAE based, state difference based (DIFF) and state transition probability based (PROBA) to detect each device: fridge (FR), coffee machine (CF), kettle (TE), microwave (MW), television (TV) and monitor (MO). | 45 |
| 3.3 | Power demand of devices in REDD dataset <i>House 1</i> | 47 |
| 4.1 | Running example of DP in SmartSense for one-state devices | 61 |
| 4.2 | Running example of DP in SmartSense for multi-state devices | 64 |
| 5.1 | List of devices in REDD 1 and their characteristics. | 72 |
| 5.2 | List of devices in REDD 2 and their characteristics. | 73 |
| 5.3 | List of devices in UK-DALE 5 and their characteristics. | 73 |
| 5.4 | List of devices in Athemium dataset and their characteristics. | 73 |
| 5.5 | Regularization parameters. | 74 |
| 5.6 | Performance of the algorithms without monitored device (%). | 74 |
| 5.7 | Performance per device in REDD 1 without monitored device (%). | 75 |
| 5.8 | Performance per device in REDD 2 without monitored devices (%). | 75 |
| 5.9 | Performance per device in UK-DALE 5 without monitored device (%). | 75 |
| 5.10 | Performance per device in Athemium without monitored device (%). | 76 |
| 5.11 | Most effective devices with the CPH and DP algorithms. | 79 |
| 5.12 | Most effective devices with DTW and ED algorithms. | 84 |
| 5.13 | Performance per device in REDD 1 (%). | 87 |
| 5.14 | Performance per device in REDD 2 (%). | 88 |
| 5.15 | Performance per device in UK-DALE 5 (%). | 88 |
| 5.16 | Performance per device in Athemium dataset (%). | 88 |
| 5.17 | Number of computations. | 89 |

Chapter 1

Introduction

1.1 Smart Grid and Smart Building Automation

Natural resource nowadays is the motivation for many researches on energy management in buildings and homes. Energy management is not only the reduction of energy consumption but also the balance between energy consumption and energy generation. Smart Grid (SG) and Smart Building Automation (SBA) are the approaches that have the main impact on energy saving issues.

The concept of SG is associated with the energy production side [23]. Concretely, the *grid* refers to the electric grid or a network of transmission lines, substations, transformers and other components that deliver the electricity from the power plants to the consumers. Meanwhile, modern energy management techniques and communication infrastructures such as Internet, wireless communications or Power-Line Communications (PLC) make the *grid* smart with the possibility of automated monitoring, computing, controlling without human intervention [24–27]. In addition, the purpose of SG also includes the integration of large-scale renewable energy systems such as wind power along the farms or solar power on the top of buildings in order to protect the environment and ecosystem, instead of using only the energy generated by the traditional power plants, e.g. hydro power, thermal power or nuclear power.

In contrast, in SBA as in Figure 1.1, the principle problem is how to reduce the energy consumption inside homes or buildings. A smart building can be achieved by the

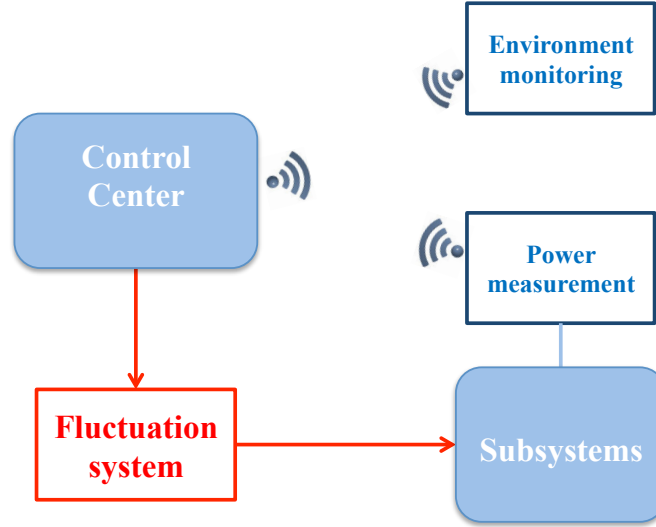


FIGURE 1.1: Principle components of a SBA system.

generation and efficient usage of energy. The modality of energy usage comes from subsystems such as lighting system, office equipment, Heating, Ventilation and Air Conditioning (HVAC), etc. Therefore, the energy waste elimination effort can concentrate on these systems by sensing and automatically fluctuating their power to adapt to the environment condition. In a smart system, human intervention must be restricted, the automation system automatically control the equipment through such a Energy Management System (EMS). Similar to SG, in SBA, the communication infrastructure plays an important role in exchanging the information among the components inside the EMS. That is the reason why PLC and Wireless Sensor Networks (WSN) can be considered as potential solutions for environment monitoring and subsystem fluctuation, but WSN is more promising because of its flexibility and low cost. With WSN, the sensors capture the environmental parameters and report data to the EMS center through wireless transceivers. The control center, with the information about the state of subsystems, then fluctuates their power consumption to adapt to the environmental variation.

The monitored environmental parameters can be occupancy, light intensity, temperature, moisture, sound, etc. Detecting and analyzing these information allows the system to automatically regulate the equipment to comfort the users, especially with the devices requiring time to start up such as HVAC system, and to regulate the power consumption. Except for occupancy, other types of environment can be easily monitored by the corresponding sensors such as light sensors, temperature sensors, humidity sensors, microphones, etc. Meanwhile, the occupancy detection is an interesting subject for many

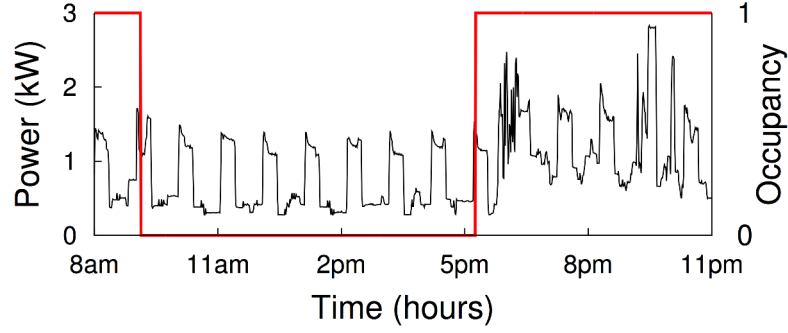


FIGURE 1.2: Variation on the power signal when the occupant turns on some devices [5].

recent researches [28–31]. The occupancy can be detected directly by a Passive Infrared Sensor (PIR) or camera [28–30], or indirectly by analyzing the events happening inside house or building, e.g. door state, variation on the total power signal [5], abnormally increase of the CO₂ concentration [32]. For example, a couple of magnetic Reed Switch (RS) door sensor and PIR sensor can be used to detect the occupancy in offices or rooms [28, 29]. Although PIR sensors are the most ubiquitous form for motion sensing, it is not enough for occupancy detection because people inside rooms/offices tend to maintain their posture when working in front of a computer, watching television or sleeping. To overcome this challenge, combining with an RS door sensor, which can determine if the door is open or closed, is more efficient. Another example of occupancy monitoring is to detect the large changes on the aggregate power signal and infer the occupancy [5]. As shown in Figure 1.2, the background load frequently appears and does not imply the occupancy. This load is derived from the presence of the devices driven by an automated controller such as fridge or HVAC system. When having the presence of the occupants, their physical interactions with loads such as switching on the lamp, turning on the television, etc., will imply changes on the power signal. The large changes can be detected by comparing the average power, standard deviation or power range in a sliding window of signal with corresponding thresholds. Obviously, the accuracy of this monitoring system strongly depends on the survey of background load due to the variation of this parameter according to the time of day and year such as day or night, spring or winter. Besides, the state transition of some types of devices can also provide the interaction of users [33]. For example, a washing machine has five states including off, wash, rinse, spin and maintenance wash, the user interaction can make the transition from the first to the second state.

In SBA, besides the environment monitoring, the power consumption of the equipment

also needs to be measured and reported so that the control center can make a suitable fluctuation to adapt to the environmental variation.

1.2 Load Monitoring

1.2.1 Intrusive Load Monitoring

Similar to the environment monitoring sensors, power measurement units also require a wireless protocol to communicate with the control center. Power metering can be implemented by two ways: intrusive and non-intrusive. In intrusive approach, the measurement is deployed at each individual equipment. In contrast, the Non-Intrusive Load Monitoring (NILM) ¹ can detect and estimate the power consumption of all devices on the monitored branch circuit with only one power meter installed on the main power entry [1]. Intuitively, non-intrusive approach helps to reduce the deployment cost but requires a long period of observation to analyze the power characteristics of each device.

As shown in Figure 1.3, an Intrusive Load Monitoring (ILM) system can be implemented in a direct or indirect method. The direct one attaches a power meter to each device. Therefore, direct ILM obviously performs with high accuracy but has some disadvantages such as high cost and interruption on the power supply to install the meters. In contrast, the indirect method uses low-cost sensors to detect the measurable signals emitted by the device to estimate the power consumption. For example, a light sensor can detect the power state of a television and infer the corresponding consumed energy, an acoustic sensor can be applied to recognize the operation of the fridge compressor [6].

1.2.2 Non-Intrusive Load Monitoring

In contrast with intrusive approaches, the non-intrusive ones rely on only one power meter at the main power supply. An NILM algorithm tries to extract some specific features from the aggregate power signal to identify the corresponding devices. Assuming there are N devices connected to a unique power meter, each device $i \in \{1, \dots, N\}$ can operate at one state j of m_i modes, which consumes a power of w_{ij} , $j \in \{1, \dots, m_i\}$. Denote $s_{ij}(t)$

¹sometimes also referred as NIALM (Non-Intrusive Appliance Load Monitoring) or Electrical Load Disaggregation

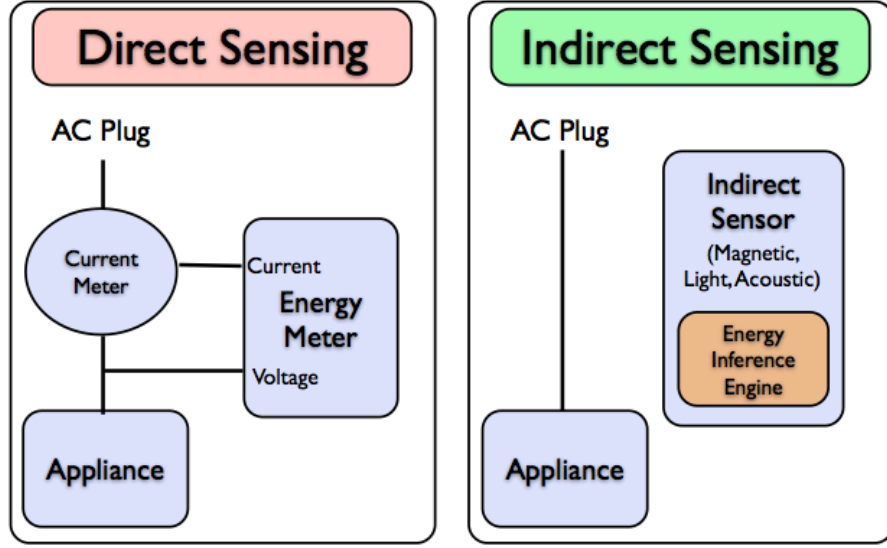


FIGURE 1.3: Direct and indirect power measurement [6].

as the Boolean indicator of state j of device i at time t :

$$s_{ij}(t) = \begin{cases} 1 & \text{if device } i \text{ operates at state } j \\ 0 & \text{otherwise.} \end{cases}$$

The aggregate power consumption can be represented as follows:

$$x(t) = \sum_{i=1}^N \sum_{j=1}^{m_i} s_{ij}(t) \times w_{ij} + e(t), \quad (1.1)$$

where $e(t)$ is a noise or error term. From this model, the vector \mathbf{s} containing all state indicators s_{ij} , $i = 1, \dots, N, j = 1, \dots, m_i$ of the devices from 1 to N can be determined by solving the following minimization problem:

$$\hat{\mathbf{s}}(t) = \arg \min_{\mathbf{s}(t)} \left\| x(t) - \sum_{i=1}^N \sum_{j=1}^{m_i} s_{ij}(t) \times w_{ij} \right\|_d, \quad (1.2)$$

where $\| \cdot \|_d$ denotes the l_d distance. In the context of NILM, we can use some distance metrics such as l_1 , l_2 . This problem is computationally intractable and impractical to be exactly solved by exhaustive techniques unless N is small. Instead, heuristic algorithms might be considered. There are three principles for any NILM algorithm, including:

- Select and mathematically characterize the specific features or signatures.
- Deploy the suitable hardware to measure the desired signal.

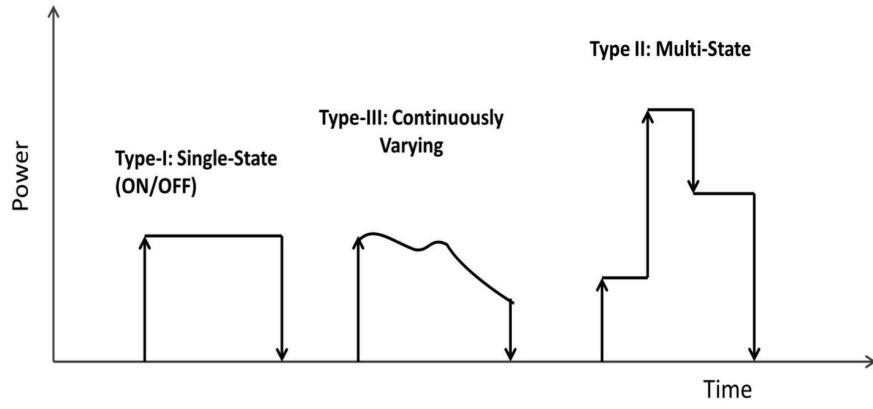


FIGURE 1.4: Different categories of load based on energy patterns [7].

- Develop an algorithm to detect those features/signatures from the signal.

The characterization of features and signatures needs the knowledge on the operation states of the devices, which can be categorized into four types, as surveyed in [7], including:

- Type-I: devices with two states (ON/OFF) such as lamp, toaster, etc.
- Type-II: finite state machines with a finite number of operation states, e.g. washing machine, stove, etc.
- Type-III: continuously varying devices with variable power consumption such as power drill or dimmer lights.
- Type-IV: devices that remain active throughout days or weeks and consume a constant level of power such as smoke detector, internet modem, etc.

The three first categories are proposed by [1] and shown in Figure 1.4, while the last one is highlighted in [34, 35]. Besides, the selection of features/signatures also depends on the sampling frequency of meters and can be divided into two groups: low frequency and high frequency. Algorithms based on low-frequency signal focus on detecting the steady-state features such as average power, step-changes, etc., while high-frequency hardware allows us to identify the devices from the transient phase and harmonics.

1.2.3 Hybrid Load Monitoring

A hybrid approach between intrusive and non-intrusive load monitoring, called Hybrid Load Monitoring (HLM), was proposed in recent years [21, 22, 36–38]. In this approach, a sensor network is deployed in homes to monitor the environmental parameters such as acoustic noise, light intensity, occupancy, etc., to detect the operation of some specific devices. The difference between the hybrid and intrusive approaches is that the sensors are only installed to monitor a part of all devices while the others are still identified by an NILM algorithm with aggregate power measurement. Therefore, HLM allows for a reduction of the cost of the intrusive approach and an increase of the detection accuracy of the non-intrusive one. In the context of this thesis, we will focus on hybrid methods in order to improve the performance of the existing NILM algorithms.

1.3 Thesis Outline and Contributions

Chapter 2: State of the art

This chapter reviews the techniques and approaches used in load monitoring, from intrusive to non-intrusive and from low frequency to high frequency sampling hardware. In intrusive approach, each device is attached to a set of sensors, which detect the environmental variation generated by the device and infer its power state. Although this approach shows a high accuracy, it is difficult to deploy in smart homes and buildings because of high cost and too much technical intervention on the infrastructure. Instead, by using only one power meter at the main entry of electricity, non-intrusive approach is more promising to study. In this approach, the electrical features of each device are characterized and an algorithm is developed to extract the features from the aggregate power load and therefore to identify the corresponding device. The selection of electrical characteristics depends on the sampling frequency of hardware. Concretely, low frequency hardware is suitable for the stable loads with the features related to the average power demand, rising step-changes, falling step-changes, etc., while high frequency hardware is more efficient to identify the devices based on the transient signal and harmonics analysis. In recent years, a hybrid approach of intrusive and non-intrusive is studied, in which the sensors can be deployed to detect the state of several devices, while the rest is still recognized by a non-intrusive algorithm.

Chapter 3: l_1 -norm minimization based algorithms for NILM

In this chapter, the minimization problem in NILM is solved by directly using the brute force method to determine the state of each device. In this method, all possible combinations of operating devices are in turn applied to calculate the absolute error between their total power demand and the measured aggregate power. The set of devices giving the minimum error are identified as running. However, if there is more than one device having the same power demand, the detection is no more accurate. The main contribution of this chapter is to propose two solutions to overcome this challenge and to improve detection accuracy, including:

- Difference-based algorithm to compare the difference between the current state and the previous state of each device.
- Probability-based algorithm to use the state transition probability to decide the operating devices.

An experiment is also implemented with a small set of devices installed our laboratory.

Chapter 4: SmartSense: Sensor-Aided NILM

The main contribution of this thesis, SmartSense system, is presented in this chapter. To improve performance of NILM algorithms, we propose to use the operating probability of each device as an external information. The probability can be estimated from a sensor network. Different from the intrusive approach, only a small subset of some devices are selected to be monitored by the sensors while the others are assumed to be equiprobable among the states. Although it could be perceived as similar to a hybrid approach, however, instead of identifying the state of devices from the sensor signals, we transform this state to a probability and an NILM algorithm can still be applied to all devices combined with this external information. Therefore, monitoring a set of devices can help to improve the detection of the others. The algorithms applied to SmartSense in this thesis include:

- Compositional Pareto-algebraic Heuristic.
- Dynamic Programming.
- Edge Detection.

- Dynamic Time Warping.

These proposed algorithms are applied to both our own dataset as well as publicly available dataset. The evaluation of performance improvement is given in **Chapter 5: Experimental results**. Meanwhile, the conclusions and perspectives of this thesis will be presented in **Chapter 6: Conclusions and perspectives**.

Chapter 2

State of the Art

2.1 Intrusive Load Monitoring

A naive, intrusive and costly method to measure the power consumption of each device is to connect it to a power meter. Apparently, this approach is practically unfeasible because of high cost and too much technical intervention on the power supply. Therefore, Kim et al. [6] propose a so-called ViridiScope system to replace part of the power meters by indirect sensors to detect the environmental parameters around the monitored devices and deduce their power state. In this monitoring system, devices with stable loads can be monitored by indirect sensors such as light intensity sensors, acoustic sensors, etc., while devices with variable loads are attached to a magnetometer or a direct power meter. A power meter is also installed at the main power line to measure the aggregate power consumption and an algorithm is developed to disaggregate this power to the corresponding devices. An overview on the power estimation based on sensors and meters is presented in the following.

Magnetometer: The magnetic field around device i is sampled at 100 Hz and the signal $s_i(t)$ is formed from the standard deviation over one second sliding window. The corresponding power consumption correlated to this signal is estimated as:

$$x_i(t) = \alpha_i s_i(t) + \beta_i, \quad (2.1)$$

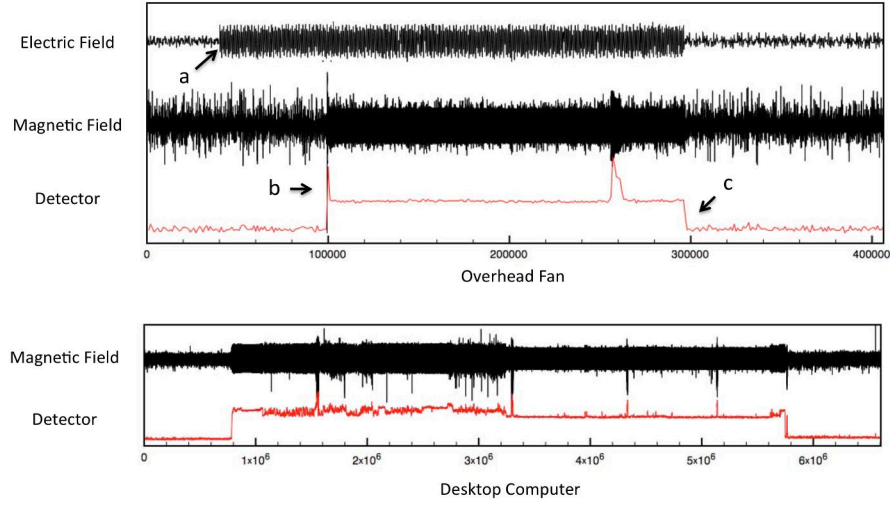


FIGURE 2.1: Electromagnetic field waveform. Top: Overhead fan is powered up at point (a), switched on at point (b) and switched off at point (c). Bottom: Variations in magnetic field near by a desktop computer [8].

where α_i, β_i are the calibration parameters. The magnetic field analysis method is also applied in [8], but combined with electric field, as illustrated in Figure 2.1. The electric field is generated from the difference of voltages, i.e. a strong electric field may be generated even when the device does not draw current, while magnetic field variations correspond to the changes in power consumption. Therefore, the electric field can be used to determine if the device is powered or not.

Environment monitoring sensors: To estimate the power consumption of the devices with a limit number of power states, environment monitoring sensors, such as light or acoustic sensors, can be deployed. Denote $s_{ij}(t)$ a boolean indicator of the internal state j of device i , the power consumption is estimated by:

$$x_i(t) = \sum_{j=1}^{M_i} w_{ij} \cdot s_{ij}(t), \quad (2.2)$$

where M_i is the number of internal states of device i and w_{ij} the average power demand of state j .

Direct power meter: There are still other devices monitored by a direct power meter and their power consumption is directly read from the meter. For device i ,

$$x_i(t) = \tilde{x}_i(t), \quad (2.3)$$

where $\tilde{x}_i(t)$ is the signal from the meter connected to device i .

Uninstrumented devices: Because of a large number of electrical loads in homes and buildings, some of them cannot be monitored by any meter or sensor. They are called uninstrumented devices. Their total power consumption can be considered as consumed by a unique device. Denote w_i as the total power demand of these devices, their power consumption is estimated from the operating state as

$$x_i(t) = w_i s_i(t), \quad (2.4)$$

where $s_i(t)$ indicates if the uninstrumented devices are present in the configuration or not.

Obviously, the total power consumption in home or building is the sum of the power consumption of all devices plus some noise and is defined as

$$x(t) = \sum_{i=1}^N x_i(t) + n(t)$$

where

$$x_i(t) = \begin{cases} \alpha_i s_i(t) + \beta_i: \text{magnetometers} \\ \sum_{j=1}^{M_i} w_{ij} s_{ij}(t): \text{light/acoustic sensors} \\ w_i s_i(t): \text{uninstrumented} \\ \tilde{x}_i(t): \text{direct meter input} \end{cases} \quad (2.5)$$

where N is the number of devices and $n(t)$ is a noise. To estimate the power demand of each device, i.e. w_{ij} if monitored by sensors and w_i if uninstrumented, as well as the calibration parameters α_i, β_i , the following numerical optimization problem formulated from equations (2.1), (2.2), (2.3), (2.4) needs to be solved:

$$\min_{\theta} \left\| x(t) - \sum_{i=1}^N x_i(t) \right\|, \quad (2.6)$$

where θ is a vector containing all variables of power demand and calibration parameters, $\| \cdot \|$ denotes the l_1 -norm or least absolute value regression solving for the median value. An example of power disaggregation in ViridiScope is represented in Figure 2.2.

Despite showing a high accuracy ($> 90\%$) in the experiment [6], ViridiScope still has some drawbacks. The first one is that the uninstrumented devices are assumed to consume a constant power, while this power can vary depending on the type of devices. The second

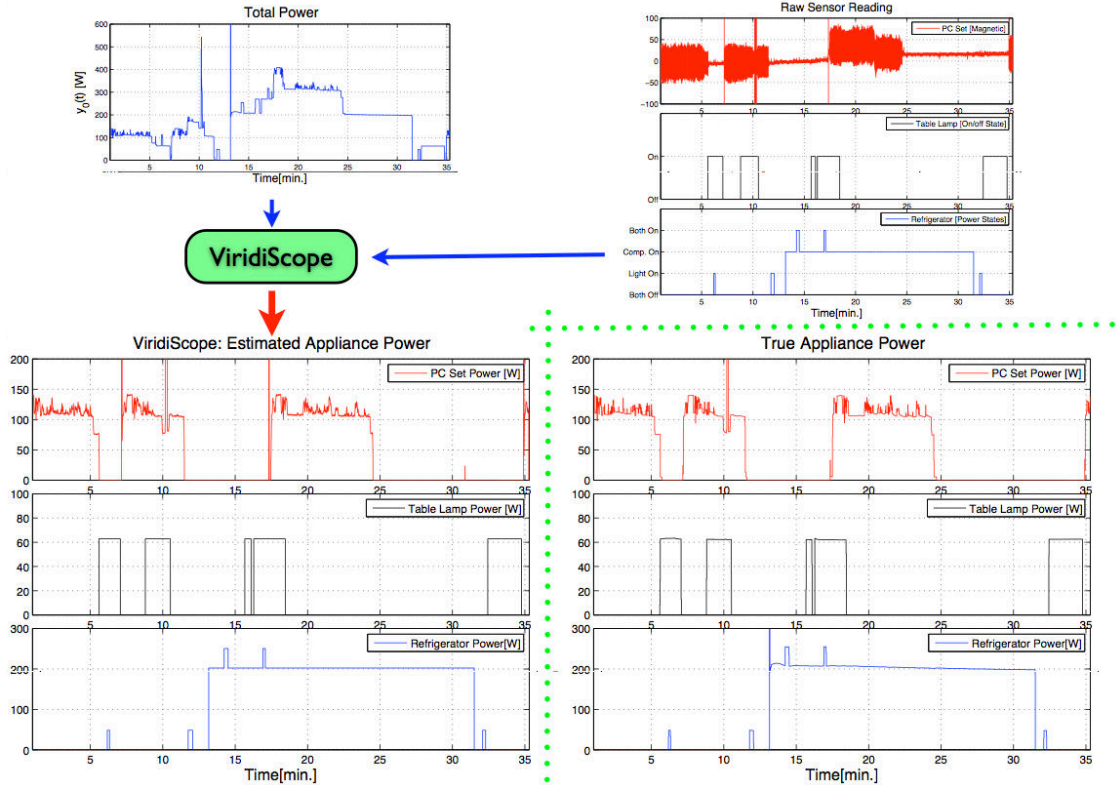


FIGURE 2.2: The ViridiScope system considers the total power consumption, magnetic field signal and internal power state information to estimate the power consumption of each device by solving an l_1 -norm minimization problem [6].

limitation relates to the large amount of sensors necessary for deployment to reduce the number of uninstrumented devices and increase the accuracy of the system.

Similar to ViridiScope, in [9, 39], the authors propose to use binary sensors to detect the on/off state of the energy consumers inside the buildings and then disaggregate the total power consumption to each one. The fundamental difference between this method and ViridiScope is to use the binary sensors providing the on/off state instead of raw data. All devices are assumed to consume a stable power level during their operation. That is the reason why the power consumption of the variable loads or multi-state devices cannot be accurately estimated. To increase the accuracy, the length of observation window used to estimate the power consumption of each device needs to be short enough. Besides, if the on/off sequences of more than two devices are synchronized, the power cannot be disaggregated. In this case, to reach a desired accuracy, some additional power meters are placed to the necessary electrical circuit branches, as illustrated in Figure 2.3. In this example, because the on/off sequences of devices x_2 and x_5 are synchronized, their power is impossible to estimate if they are connected to the same power meter as in the

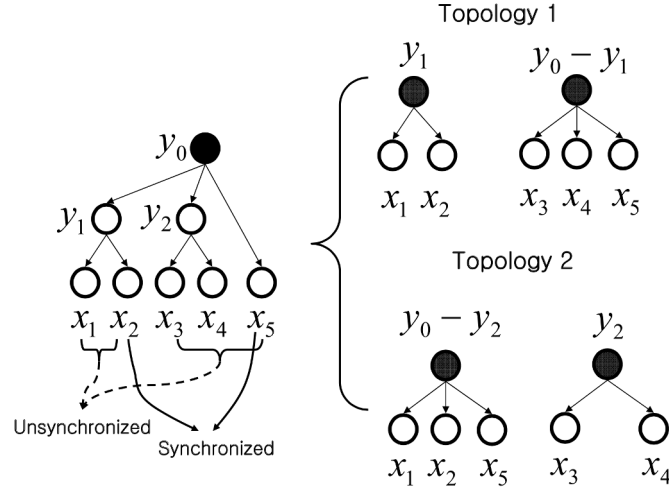


FIGURE 2.3: Power meter topology. The left topology uses only one power meter (dark node) at the main power supply while topology 1 and 2 use two additional power meters at the electrical circuit branches [9].

left topology and topology 2. In contrast, in topology 1, these devices are connected to different circuit power meters, the disaggregation process can give a better performance.

A shortcoming of the research in [9, 39] is to ignore the presence of the *ghost power* coming from the uninstrumented devices, that has strong effect on the performance when Beckel et al. [40] apply this method to the Reference Energy Disaggregation Dataset (REDD) [4]. To overcome this problem, they assume to add an always-on *virtual ghost power consumer* to the set of monitored devices. They also assume that all devices are perfectly monitored by the binary sensors. By this way, the disaggregation can play a better performance, defined by the difference between the estimated power and real power consumption of each device.

2.2 Non-Intrusive Load Monitoring

Although direct power meters can be replaced by indirect sensors to reduce the deployment cost [6, 9, 39], a large amount of sensors as well as technical intervention is still required. To radically decrease the implementation cost, the presence of sensors needs to be eliminated or reduced as much as possible and the state of devices needs to be determined based on the aggregate power measured at the main power supply. This approach is called Non-Intrusive Load Monitoring or NILM and firstly proposed by G. Hart [1] in the early 1990s. In recent years, there are several industrial solutions hit to the market

such as Wattseeker [41], Smart Analyzer [42]. In NILM, an algorithm is developed to detect some specific features to identify the operating devices. The features are selected depending on the sampling frequency of the meters and can be classified into two classes: low frequency and high frequency.

2.2.1 Low frequency approaches

In the low frequency approach, the promising features relate to the real and reactive power such as average power and step-changes when switching on/off or changing the power state of a device [1, 43]. The step-change analysis method was firstly introduced by G.Hart [1], in which the large changes on the power signal, as shown in Figure 2.4, are detected by an edge detector. The edge detector firstly segments the power signal into steady periods, defined as a set of consecutive instants in which the power does not vary greater than a threshold. The samples in each period are then averaged and the difference between two consecutive steady periods is called step-size. Denote $x(t)$ as the measured power at time instant t , $x(t)$ is then expressed as

$$x(t) = \sum_{i=1}^N x_i(t) + n(t), \quad (2.7)$$

where N is the number of devices, $x_i(t) \geq 0$ is the power consumption of device i and $n(t)$ is a noise at instant t . A step-change is detected if $|x(t) - x(t-1)| \geq \gamma$, with threshold γ . An event is determined as started at time t_s and ended at time t_e if:

$$|(x(t_s) - x(t_s - 1)) + (x(t_e) - x(t_e - 1))| \leq \alpha, \quad (2.8)$$

where α is a practical estimated parameter. A cluster analysis maps the detected step-changes to a two-dimensional space of real and reactive power. The rising edge (positive step-change) and falling edge (negative step-change) of an event (device activation) are then paired together and compared with the existing clusters from the training procedure to identify the corresponding load using maximum likelihood algorithm. Figure 2.5 shows the position of some devices on the P-Q plane.

The edge detector based method proposed in [1] is suitable to detect and identify the on/off events on the power signal. However, it cannot detect the variable loads such as computers, whose power consumption depends on the tasks. In addition, as illustrated

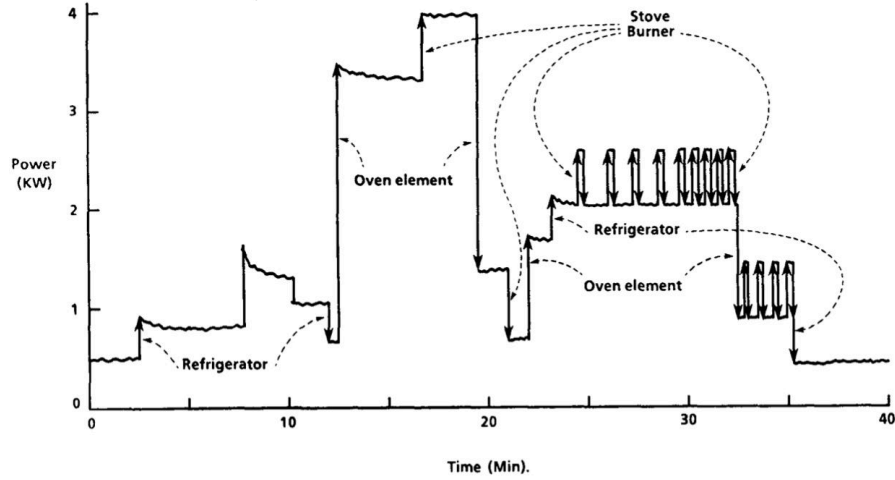


FIGURE 2.4: Step-changes on the aggregate power signal [1].

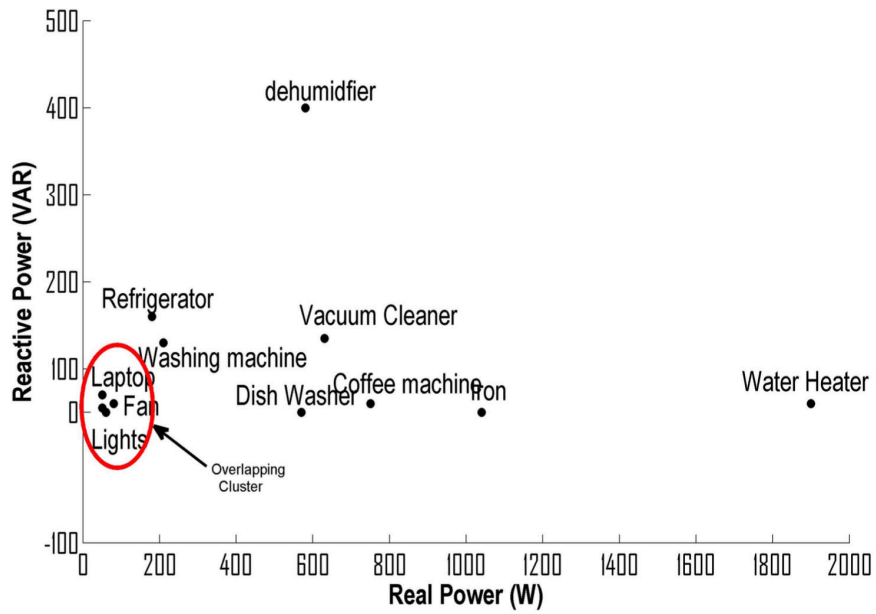


FIGURE 2.5: Load distribution in P-Q plane [10].

in Figure 2.5, the devices with overlapped clusters on the P-Q plane cannot be discriminated.

To improve the performance of edge detection, some later researches are developed with some additional features. For example, in [44, 45], a median filter is applied to remove the meaningless abrupt peaks in the raw signal, while in [46], edges and slopes, defined as the slow variations after an initial spike in the power signal when switching on a device, are simultaneously used as a feature to detect the devices necessary for a starting time before reaching the steady state such as heat pumps, dish washers, fridges, etc.

Similar to [1], the authors of [2] try to detect and pair the rising and falling edges of an event and construct a decision tree to identify the corresponding device. In the training phase, for each device, only edges with maximum and minimum height are saved instead of all possible ones. Thus, there are only $2N$ values in the library. The decision tree includes a *root*, several *nodes* and *leaves* relating to known or undefined devices. At each node, a split point value $V(\text{node})$ calculated from the training data is used to decide the *child* node, i.e. left or right. The splitting procedure iterates until any *leaf* node is reached, i.e. the event is matched to a corresponding device.

Besides decision tree, another so-called Dynamic Time Warping (DTW) algorithm is also presented in [2]. In this method, all active power values of an event from the rising edge to the falling edge are extracted and saved in a vector to create a pattern instead of only two edges. Because the lengths of the events are different, a pattern matching procedure via dynamic programming is proposed to apply. When a pattern is detected, the accumulated distance between it and all existing patterns in the library is calculated by Algorithm 1. The pattern giving the smallest distance is selected to identify the corresponding device.

Algorithm 1 Accumulated distance between two vectors with different lengths [2].

```

1: function DTW( $p, q$ )
2:    $n = \text{length}(p)$ 
3:    $m = \text{length}(q)$ 
4:    $D(0, 0) = 0, D(i, 0) = D(0, i) = \infty$ 
5:   for  $i = 1, \dots, n$  do
6:     for  $j = 1, \dots, m$  do
7:        $d(i, j) = |p(i) - q(j)|$ 
8:        $D(i, j) = d(i, j) + \min \{D(i-1, j), D(i-1, j-1), D(i, j-1)\}$ 
9:     end for
10:  end for
11:  end for
12:  end for
13:  Output  $D(n, m)$ 
14: end function
15: end function

```

The simulation results with REDD dataset [2], as shown in Table 2.1, point out that the DTW algorithm outperforms the Hidden Markov Model (HMM) approach ¹ in terms of detection reliability as well as sensibility to the events.

The active power values can also be used to compute the load distribution [47]. Whenever an event is detected on the aggregate power signal, the corresponding load distribution

¹presented later in this chapter

TABLE 2.1: Performance comparison between DTW and HMM approaches [2].

| | Reliability (%) | | Sensibility (%) | |
|---------|-----------------|-------|-----------------|-------|
| House | DTW | HMM | DTW | HMM |
| House 1 | 85.01 | 79.29 | 79.72 | 62.72 |
| House 2 | 89.86 | 51.80 | 84.39 | 62.79 |
| House 6 | 98.86 | 99.30 | 81.22 | 74.74 |

will be calculated and compared with the library to identify the device. To get the load distribution when two or more devices operate at the same time, the convolution is applied to the distribution of the corresponding devices. Meanwhile, the authors of [48] apply the edge detector to each phase of electric power supply. This division helps to reduce the state space from K^N down to K^{N_p} if each device has K states and each phase supplies the power for N_p devices. Obviously, this method is efficient when applied in large buildings with more than one phase of electric power.

In another research, to identify large loads such as water heaters, the authors of [49] propose a so-called *top-bottom rule-based* algorithm comprised of many decision rules applied to each detected pattern such as step-change, average power, total power, number of data points, etc. After comparing each component with the library, a final score is calculated to make a final decision about the operation of the corresponding device. Each type of devices is recognized by a different set of rules, which implies that this method need an excessive training and makes it difficult to be widely implemented.

Instead of detecting the events on the aggregate power signal, the raw power data can also be directly used in sparse coding [50, 51]. In this method, the training data of each device is contained in matrix $\mathbf{W}_i \in \mathbb{R}^{T \times m}$ in which each column corresponds to the data of one week. The aggregate power consumption is then $\mathbf{W} = \sum_{i=1}^N \mathbf{W}_i$. Applying the sparse coding, the following approximation is considered:

$$\mathbf{W}_i = \mathbf{B}_i \times \mathbf{A}_i, \quad (2.9)$$

where $\mathbf{B}_i \in \mathbb{R}^{T \times n}$ contains n basis functions or *dictionary*, and $\mathbf{A}_i \in \mathbb{R}^{n \times m}$ contains the *activations* of basis functions. In the training period, the values of \mathbf{B}_i and \mathbf{A}_i are estimated by the following minimization:

$$\min_{\mathbf{A}_i \geq 0, \mathbf{B}_i \geq 0} \frac{1}{2} \|\mathbf{W}_i - \mathbf{B}_i \times \mathbf{A}_i\|_F^2 + \lambda \sum_{p,q} (\mathbf{A}_i)_{pq} \text{ subject to } \|b_i^{(j)}\|_2 \leq 1, j = 1, \dots, n, \quad (2.10)$$

where $\|\mathbf{Y}\|_F \equiv (\sum_{p,q} \mathbf{Y}_{pq})^{1/2}$ denotes the Frobenius norm, and $\|y\|_2 \equiv (\sum_p y_p^2)^{1/2}$ is the l_2 -norm. To disaggregate a new set of data $\mathbf{X} \in \mathbb{R}^{T \times m'}$, the value of \mathbf{A}_i is recalculated as

$$\begin{aligned} \hat{\mathbf{A}}_{1:N} &= \arg \min_{\mathbf{A}_{1:N} \geq 0} \left\| \mathbf{X} - [\mathbf{B}_1 \cdots \mathbf{B}_N] \begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_N \end{bmatrix} \right\|_F^2 + \lambda \sum_{i,p,q} (\mathbf{A}_i)_{pq} \\ &\equiv \arg \min_{\mathbf{A}_{1:N} \geq 0} F(\mathbf{X}, \mathbf{B}_{1:N}, \mathbf{A}_{1:N}). \end{aligned} \quad (2.11)$$

Finally, the power is disaggregated to each device by

$$\hat{\mathbf{X}}_i = \mathbf{B}_i \times \hat{\mathbf{A}}_i. \quad (2.12)$$

Obviously, the sparse coding based disaggregation algorithm needs an excessive amount of training data. Additionally, this method is only used in off-line mode and not suitable for the real-time detection.

To recognize the large loads as in [49], Prudenzi [11] applies Artificial Neural Networks (ANN) to determine the time of use of three different loads from two groups including water heater (group H) and washing machine/dish washer (group W), as illustrated in Figure 2.6. In the preprocessing stage, the daily data is sampled every 15 minutes and segmented into six 8-hour patterns with 4-hour overlapped interval. Then, to classify these patterns into two clusters with and without power activation (cluster P and cluster A, respectively), three parameters including maximum power, average power and maximum power change in each pattern are applied to a Self-Organizing Map (SOM). The patterns in cluster P are then sent to the identification stage comprising two supervised Back-Propagation neural Networks (BPN) with 32 inputs, 24 hidden neurons and 32 outputs. The first net (BPN1) tries to detect the instants with energy usage in each pattern and fills in a 6×32 -cell output table. After that, the output is filtered by multiplying by the output of the preprocessing stage and sent to the second net (BPN2), where the patterns of group H will be discriminated from group W by applying the moving average operation. The output of this net will only contain the power usage indicators of the water heater. Combining the outputs of two neuron nets, the time of use of both groups will be determined in the post-processing stage. Although this research only considered two groups of devices, it contributes a new promising approach for pattern identification by applying ANN.

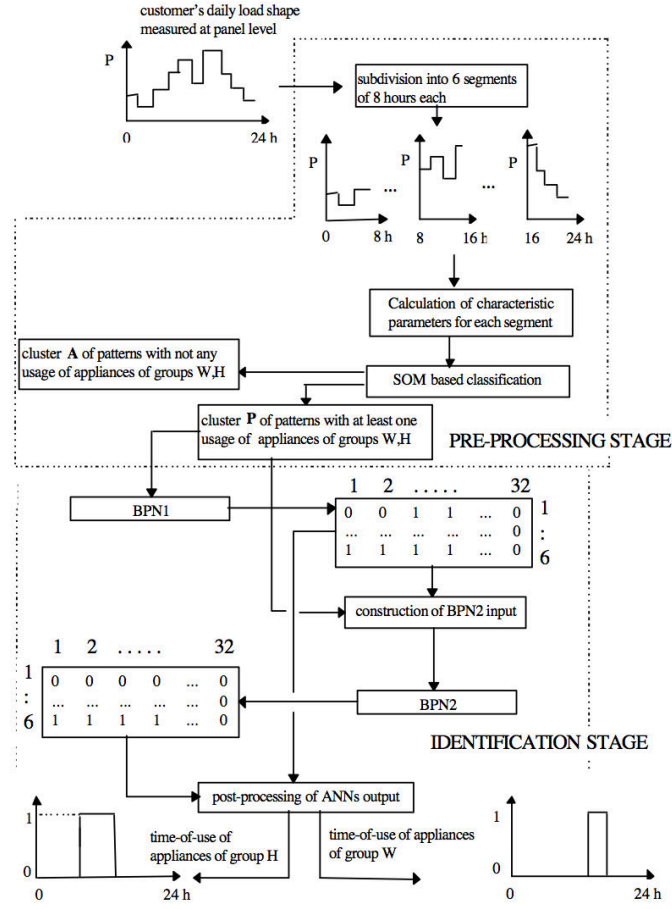


FIGURE 2.6: ANN to identify the water heater (group H) and washing machine/dish washer (group W) [11].

Similarly, an ANN architecture is also applied in a system called real-time RECOgnition of electrical Appliances and Profiling (RECAP) [52–54], where the current and voltage waveforms are analyzed to extract the peak and Root Mean Square (RMS) values of both current and voltage, phase difference and power factor to define a profile of device. Besides, two other factors including signature length and sampling frequency are also captured to standardize the signatures from dissimilar types of energy meters. Although the RECAP system can profile and recognize the devices under a single framework and show a good performance when identifying the kitchen devices, it is not suitable for the multi-state devices.

In another research, deep neural networks are constructed to disaggregate the domestic energy [12] and applied to the UK Domestic Appliance-Level Electricity (UK-DALE) dataset [55]. The input of the network is a window of aggregate data with length determined by the maximum operating duration of the target device, while the output contains the information to reconstruct its power consumption. Before to be fed to the

network, the input sequence in each data window is normalized by subtracting by the mean of sequence and by dividing by the standard deviation. The device activations used as the targets in training the networks are extracted based on some arguments such as maximum power, on-power threshold, minimum on-duration, minimum off-duration [56]. The authors of [12] propose three deep learning architectures to reconstruct the power consumption of target devices. The first one is a Recurrent Neural Network (RNN), which allows the output from each neuron in a layer at time step t to be fed via weighted connections to all neurons of that layer at time step $(t + 1)$. This makes RNNs suitable for the sequential data. To enhance the performance of RNNs in term of memory, the Long Short-Term Memory (LSTM) architecture [57], using the memory cells with a gated input, gated output and gated feedback loop, is applied. Besides, a bidirectional RNN, which is composed of two parallel RNNs, one for the input sequence forwards and one for the input sequence backwards, is also applied. The concrete architecture with type of each layer is as follows: Input \rightarrow CONVolutional layer (CONV) \rightarrow Bidirectional RNN layer (BiRNN) \rightarrow BiRNN \rightarrow Fully Connected layer (FC) \rightarrow FC (Output). Examples of neural networks using CONV, BiRNN and FC are given in Figure 2.7.

The second deep learning architecture is Denoising Autoencoder (DAE) [58]. The role of this architecture is to recover the power consumption of the target devices from the background noise. To do that, the DAEs at first try to encode the input to a compact vector with a coding layer and then decode it to reconstruct the input. The proposed architecture of DAEs includes six layers, in which the first convolutional layer can be considered as an encoder and the last one is a decoder, as follows: Input \rightarrow CONV \rightarrow FC \rightarrow FC \rightarrow FC \rightarrow CONV (Output).

Meanwhile, in the last architecture, the power activation in each data window can be reconstructed by detecting the start time, end time and average power consumption. In other words, they want to draw a rectangle around each activation in which the left side is the start time, right side is the end time and the height is the average power. Therefore, the output layer of this network includes three neurons and the output data are encoded to be in the range $[0, 1]$. The exact architecture consists of eight layers: Input \rightarrow CONV \rightarrow CONV \rightarrow FC \rightarrow FC \rightarrow FC \rightarrow FC \rightarrow FC (Output).

Figure 2.8 illustrates the outputs produced by three deep neural networks in [12]. Apparently, the DAE architecture performs a good estimation for the varying power consumers

such as washing machine, while the third architecture (Rectangles) is suitable for the stable loads such as kettle, fridge. In addition, the RNN using LSTM architecture gives a worse estimation than DAE but better than Rectangles with variable loads.

Besides ANN, HMM is also applied to find the state of devices in NILM [4, 13, 14, 59–63]. The state of each device can be considered as a hidden variable, while the observations relate to the aggregate power consumption. In [13], the aggregate power consumption x as well as the difference on the power between two consecutive time slices, denoted as $y = x_i - x_{i-1}$, are considered as the observations of a so-called difference HMM, as represented in Figure 2.9, to find the state sequence s of the devices. The correlation of the variables in this model are defined by a set θ of three parameters learned from the training dataset

$$\theta = \{\pi, \mathbf{A}, \phi\}, \quad (2.13)$$

where π denotes the probability of the initial state, \mathbf{A} the transition probability matrix from a state at time $(t-1)$ to another at time t and ϕ the emission probability generated by a state and assumed to follow the Gaussian distribution. Besides, a constraint that a device is only in on-state if the aggregate power is larger than its own power demand is also necessary to be imposed. The state of each device is determined by applying the extension of Viterbi algorithm through two steps. In the first step, a filter is applied to suppress the observations whose joint probability does not exceed the predefined threshold and the second step evaluates the joint probability of all sequences x, y, s to identify the state of devices is defined as

$$p(x, y, s|\theta) = p(s_1|\pi) \prod_{t=2}^T p(s_t|s_{t-1}, \mathbf{A}) \prod_{t=1}^T P(w_{s_t} \leq x_t|s_t, \phi) \prod_{t \in S} p(y_t|s_t, s_{t-1}, \phi), \quad (2.14)$$

where S is the set of filtered observations and T is the length of observation sequence.

Meanwhile, in [14], only the aggregate power consumption is considered as observation sequence. In this research, the on-duration of home devices can be modeled as following the gamma distribution with a couple of parameters $\kappa^{(i)}, \theta^{(i)}$ for each device i . This assumption is drawn from the on-duration histogram of some popular electrical devices such as television, laptop, monitor, etc. as shown in Figure 2.10. Moreover, some additional features such as time of day, day of week and environment information from a monitoring sensors network are also taken into account. Considering K additional conditions with

corresponding sequences $c^{(1)}, \dots, c^{(K)}$ in HMM creates a Conditional Factorial Hidden Semi-Markov Model (CFHSMM) and is illustrated by the graphical model in Figure 2.11.

The hidden variables s^* of this model can be found by applying the Maximum Likelihood Estimation (MLE) after training the set of parameters λ ,

$$s^* = \arg \max_s P(X, s | \lambda), \quad (2.15)$$

where $P(X, s | \lambda)$ is the joint probability computed by the product of the initial probability $\psi_{in}(X, s | \lambda)$, the emission probability $\psi_e(X, s | \lambda)$, and the transition probability $\psi_t(X, s | \lambda)$, i.e.,

$$P(X, s | \lambda) = \psi_{in}(X, s | \lambda) \times \psi_e(X, s | \lambda) \times \psi_t(X, s | \lambda). \quad (2.16)$$

The difference of this model compared to other HMMs is the consideration of external information in estimating the transition probability, as follows:

$$\begin{aligned} \psi_t(X, s | \lambda) = & \prod_{i=1}^N \prod_{t: s_t^{(i)}=0} \left(\left(\prod_{j=1}^N m_{s_{t+1}^{(i)} j s_t^{(j)}}^{(i)} \right) \left(\prod_{j=1}^K f_{s_{t+1}^{(i)} j c_t^{(j)}}^{(i)} \right) \right) \\ & \prod_{t: s_t^{(i)}=1} \left(\left(\prod_{j=1: i \neq j}^N m_{s_{t+1}^{(i)} j s_t^{(j)}}^{(i)} \right) \left(\prod_{j=1}^K f_{s_{t+1}^{(i)} j c_t^{(j)}}^{(i)} \right) \right) \\ & \prod_{t: s_t^{(i)}=1, s_{t-1}^{(i)}=0} P(d = l_t^{(i)} | \kappa^{(i)}, \theta^{(i)}), \end{aligned} \quad (2.17)$$

where $m_{kjl}^{(i)} = P(s_t^{(i)} = k | s_{t-1}^{(j)} = l)$ is the conditional probability for device j of state l , $f_{kjl}^{(i)} = P(s_t^{(i)} = k | c_{t-1}^{(j)} = l)$ is the conditional probability for external feature j of value l , d is the on-duration length and $l_t^{(i)}$ is the length of on-duration of device i beginning from time t .

Another less complex model of CFHSMM also applied in NILM is the Factorial Hidden Markov Model (FHMM) [4, 59–62], in which the states of each device are considered as a hidden sequence and the external information is ignored. In those researches, we can find a Nonparametric Factorial Hidden Markov Model [62], which helps to eliminate the need of sub-metered data to train the model or the prior knowledge about the number of devices. Nevertheless, detecting the devices with many levels of power demand is a challenge. Constructing the problem in the context of NILM as an HMM can give an effective state

determination approach for low-frequency power measurement. It helps to improve the accuracy and opens many perspectives to develop in future work. The accuracy of this method varies depending on the type and number of devices. Nevertheless, this approach has some limitations such as excessive training period to consider the distribution of the variables as well as complexity computation.

2.2.2 High frequency approaches

In many cases, low-frequency based algorithms cannot discern the devices with the same power characteristics such as power demand, operating duration, rising and falling edges, etc. Besides, if a device consumes a variable energy over time, the steady state does not exist and the detection algorithm cannot accurately execute. Therefore, studies on other algorithms based on a higher sampling frequency are necessary. These algorithms will focus on harmonics and especially transient phase, which contains much useful information about the devices.

In [15, 64–68], the authors propose to use the Fourier transform to determine the current harmonics. In [15], the indistinguishable problem of the devices consuming the same power in [1] is solved by using the third harmonic obtained from the phase-locked short-time Fourier transform of current waveform collected at 8000 Hz and higher sampling frequency. As shown in Figure 2.12, an incandescent light bulb and a computer cannot be discriminated by the first harmonics, i.e. real and reactive power, but can be distinguished by analyzing the third harmonics. Additionally, as presented in [15], the operating devices can also be detected by matching the events on the transient phase with the predefined signatures obtained from the training period. For example, the transient phases of a computer and an incandescent lamp are distinguishable because heating a lamp filament is different from charging the power supply or the batteries of a computer. Furthermore, transient-based recognition also allows near-real-time identification of energy consumers. That is the reason why the transient signal is studied in many other researches [16, 69–77].

In [69, 70], short-time Fourier transform is used to compute the spectral envelope coefficients of the transient signal when starting a device. The electrical load transients detected by the system will be classified by comparing with the existing transient signatures in the library. In contrast, the authors of [71–73] try to apply an ANN to process

the transient signal. In [71], the transient period is segmented into windows and a wavelet transform is applied to each one. Meanwhile, the load current acquisition can be measured instead of power consumption to reduce the complexity and computation [72]. ANN can also be combined with a genetic algorithm to identify load demands and improve recognition accuracy of NILM results [73].

However, detecting complete transient shapes is sometimes a waste of memory and not necessary. That is the reason why in [16], the authors introduce a new time pattern called *v-shape*, which is the periods containing the significant changes during the transient phase. Figure 2.13 shows an example of the *v-shapes* in the transient signal of an induction motor. The principle of *v-shape*-based identification is to search for the precise time pattern of *v-shape* and compare with the existing patterns in the library.

A limitation of the transient-based algorithms relates to the simultaneous operation in which two or more devices are switched on at the same time. To overcome this restriction, in [17], Fast Fourier Transform (FFT) is applied to analyze the harmonics of the incoming current waveform. Both real and imaginary parts of the harmonics are the inputs of an ANN, in which the number of inputs depends on the effectiveness of the additional harmonics, while the number of outputs relates to the number of classes, i.e. number of known devices. Figure 2.14 represents the mean value and standard deviation of the harmonics of some typical devices.

Besides analyzing the spectral and harmonic content, a novel method using V-I trajectory to categorize the groups of devices is also proposed in [18, 78]. The V-I trajectory is plotted for each device using the normalized current and voltage values. Figure 2.15 illustrates an example of the voltage and current waveforms of a television and the corresponding V-I trajectory. This method can form a high performance due to the distinctive V-I curves of devices. However, it is sensitive to the operation scenario of the multi-state loads. Moreover, the trajectory patterns of small loads are difficult to be distinguished.

In another research, the authors of [19] introduce a new approach to identify the running devices by detecting the noise created on the monitored power line, based on a high frequency hardware and 2048-point FFT. In homes and buildings, there are some devices running on background and their power waveform is detected and saved on the memory. Whenever a new device is turned on, a noise is created on the background and

the respective new device can be identified by analyzing the frequency components of the measured signal. For example, Figure 2.16 shows the new signal appearing on the background noise and its extracted frequency components.

Similarly, a system called Detecting Operating States of Electronic devices (DOSE) [20] tries to detect the variation on the electromagnetic field to identify the corresponding devices. The Electromagnetic Interference (EMI), generated when an electronic device operates, is coupled to the power line and its form is picked up by a sensing hardware attached to the outlet. As shown in Figure 2.17, the time-varying EMI of a laptop is different and distinguishable at idle, medium load and high load.

Although high-frequency based methods show a perspective to significantly improve device identification, the most important drawback preventing it from being widely implemented is the deployment cost, which increases along with the frequency of hardware.

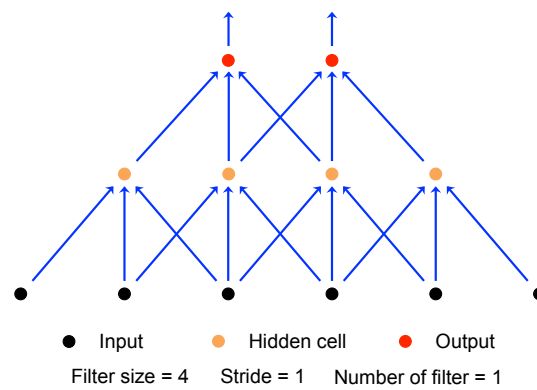
2.3 Hybrid Load Monitoring

Instead of deploying a WSN to monitor all devices in home as in ILM, a hybrid system uses less sensors to detect the environmental parameters and infer the state of some specific devices, while the others are still identified by an NILM algorithm. In order to reduce the computational complexity while still ensuring the accuracy for load monitoring, in [21], the occupancy is considered as a significant external feature because a part of devices operates during occupied periods. Figure 2.18 illustrates the correlation between the operation of some devices and the occupancy in home. Therefore, the NILM algorithms can be applied only when the occupancy monitoring system, as introduced in Chapter 1, detects an occupancy. By cutting out the unoccupied periods, the computational complexity can significantly decrease, particularly if the unoccupied periods are dominating. Obviously, the detection accuracy also decreases if there are too many background loads or automatically fluctuated devices such as fridge, washing machine, dish washer, heater, etc. Meanwhile, in [22, 36, 37], the environmental parameters such as light intensity or audio signal are analyzed to give conclusion on the operation of some specific devices, e.g. the variations in acoustic signal relate to the operation of an air blower and a refrigerator as shown in Figure 2.19. Instead of installing new sensors, the embedded sensors of mobile phones can also be used to fingerprint the profile of each

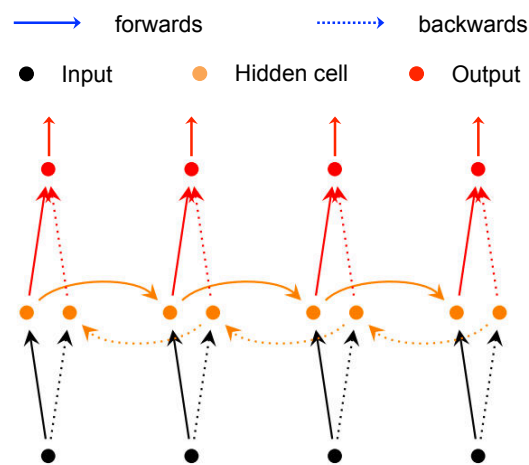
device [38]. Although the hybrid approach allows to increase the performance of load monitoring systems, more algorithms must be studied for sensors signal processing.

2.4 Conclusions

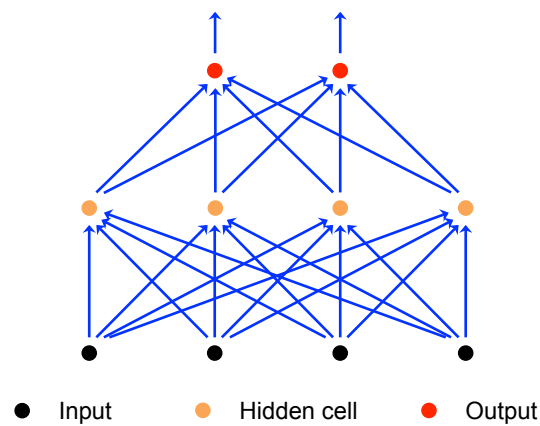
In this chapter, we have an overview on the load monitoring including intrusive, non-intrusive and hybrid approaches. ILM is the most accurate approach, but it needs very large amount of sensors to monitor all electrical consumers and therefore it is impossible to be widely applied. Meanwhile, non-intrusive approaches help to significantly reduce the deployment cost as well as technical intervention on the power supply. Nevertheless, ambiguity in the power characteristics of the devices is the most important problem in an NILM system and prevents the system from increasing the performance. As a consequence, HLM can be considered as a compromise between two approaches, which can reduce the deployment cost of ILM and increase the performance of NILM. In this thesis, we will study on hybrid methods, which use external information such as state transition probability or operating probability of each device to improve the performance of existing NILM algorithms. Concretely, in Chapter 3, we propose to use the state transition of each device from previous instant to current instant as an external information and combine it to the minimization problem in Eq. (1.2). Meanwhile, in Chapter 4, we introduce SmartSense model, in which a WSN is deployed to monitor the operation of a subset of all devices. Sensor detection is then transformed to the operating probability of corresponding devices. Disaggregation algorithms use this probability along with other specific features extracted from the global power signal to determine the state of each device.



(a) Convolutional neural network



(b) Bidirectional RNN



(c) Fully connected neural network

FIGURE 2.7: Examples of convolutional neural network, bidirectional RNN and fully connected neural network.

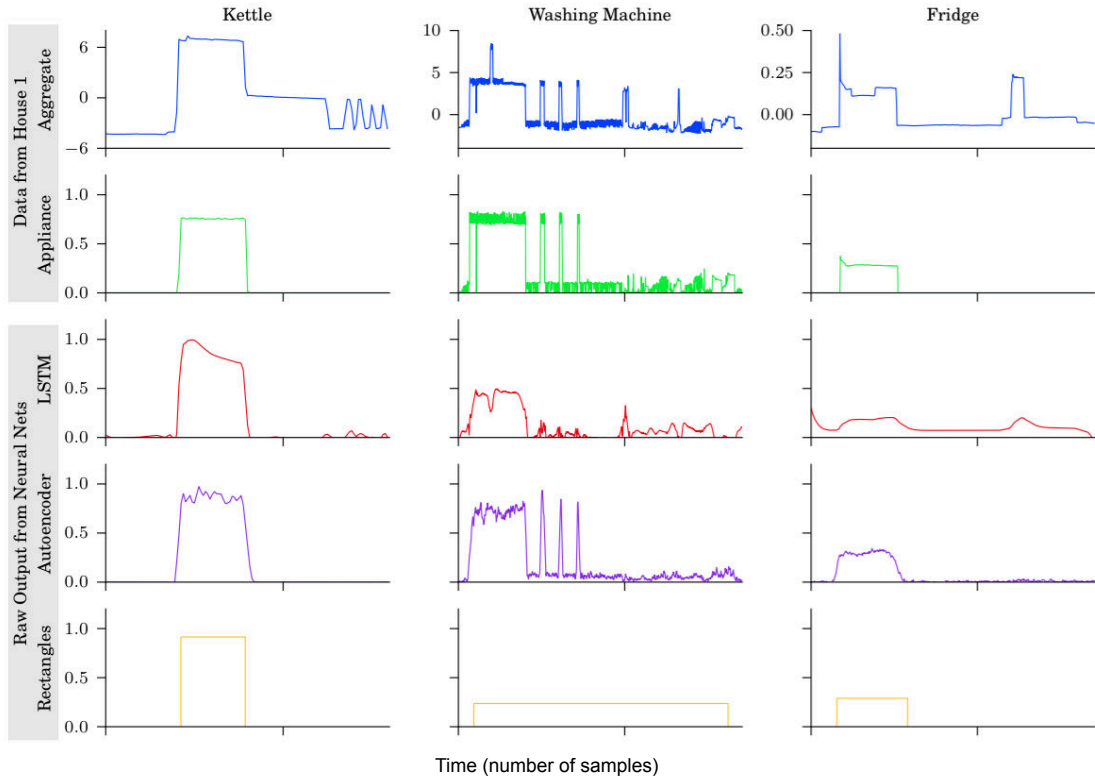


FIGURE 2.8: Outputs produced by three deep neural networks [12].

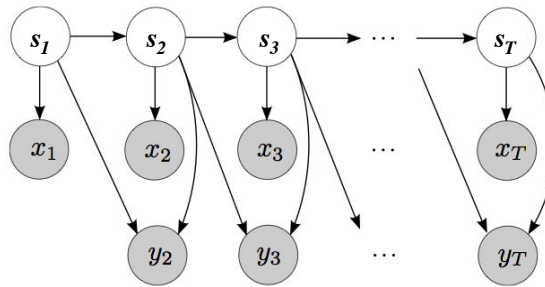


FIGURE 2.9: Difference HMM [13]. Shaded nodes represent the observations and unshaded nodes represent the hidden variables.

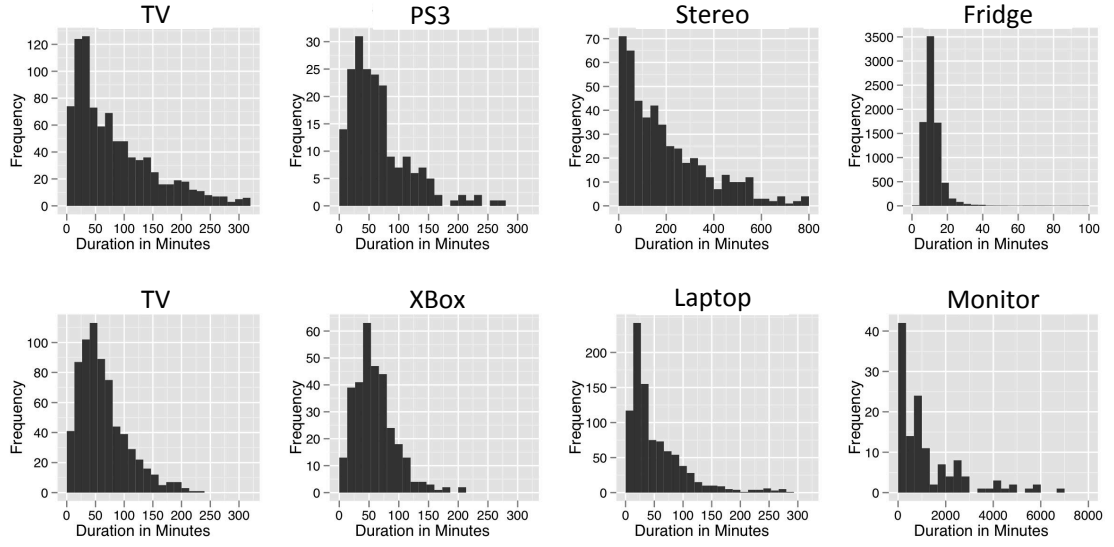
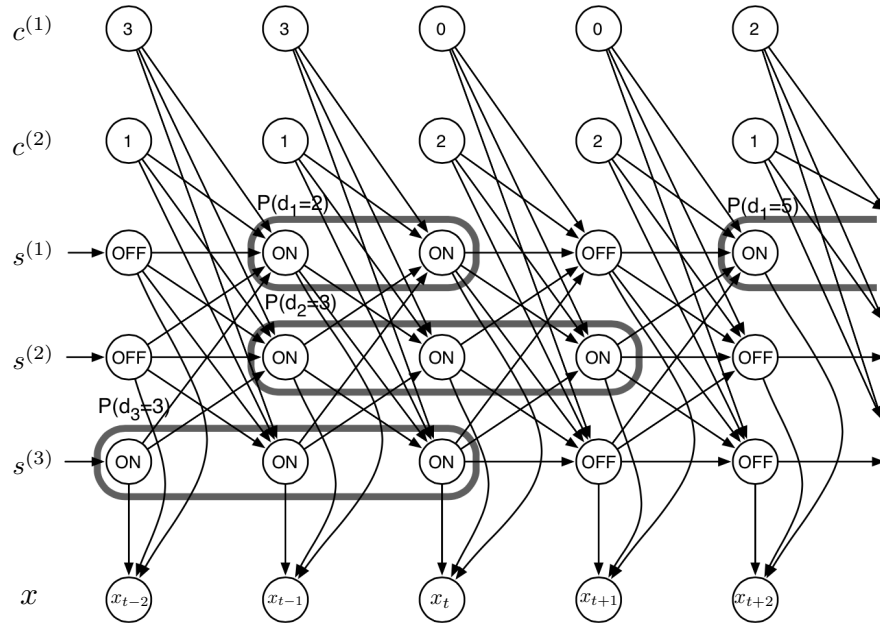


FIGURE 2.10: On-duration histogram of some devices [14].

FIGURE 2.11: Graphical representation of CFHSM with the additional conditions $c^{(1)}, c^{(2)}$ [14].

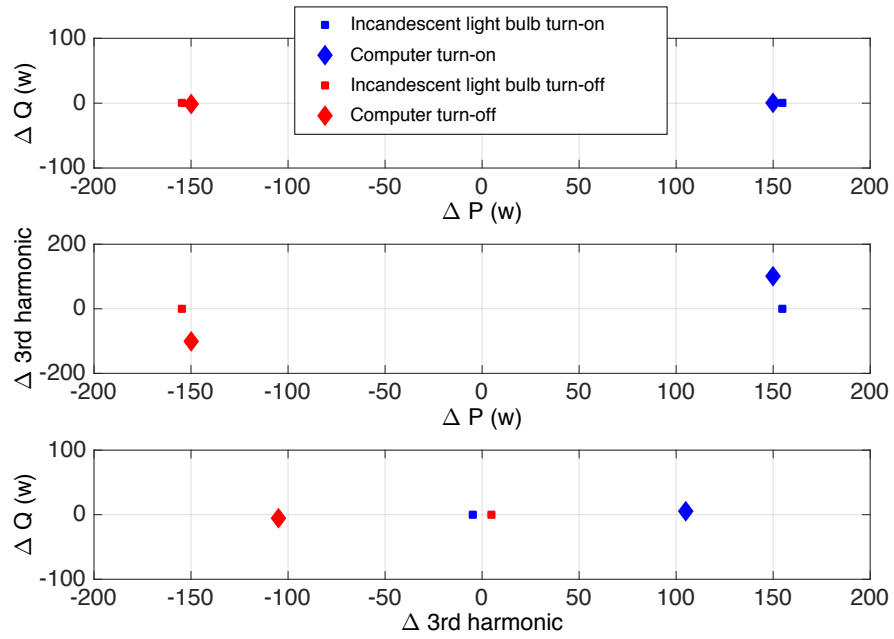


FIGURE 2.12: A computer and an incandescent light bulb cannot be discriminated by the real and reactive power but can be distinguished by analyzing the third harmonic [15].

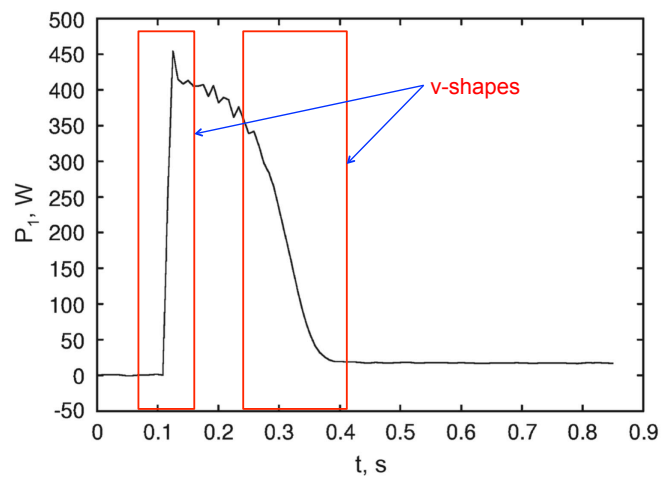


FIGURE 2.13: *v-shapes* on the transient of an induction motor [16].

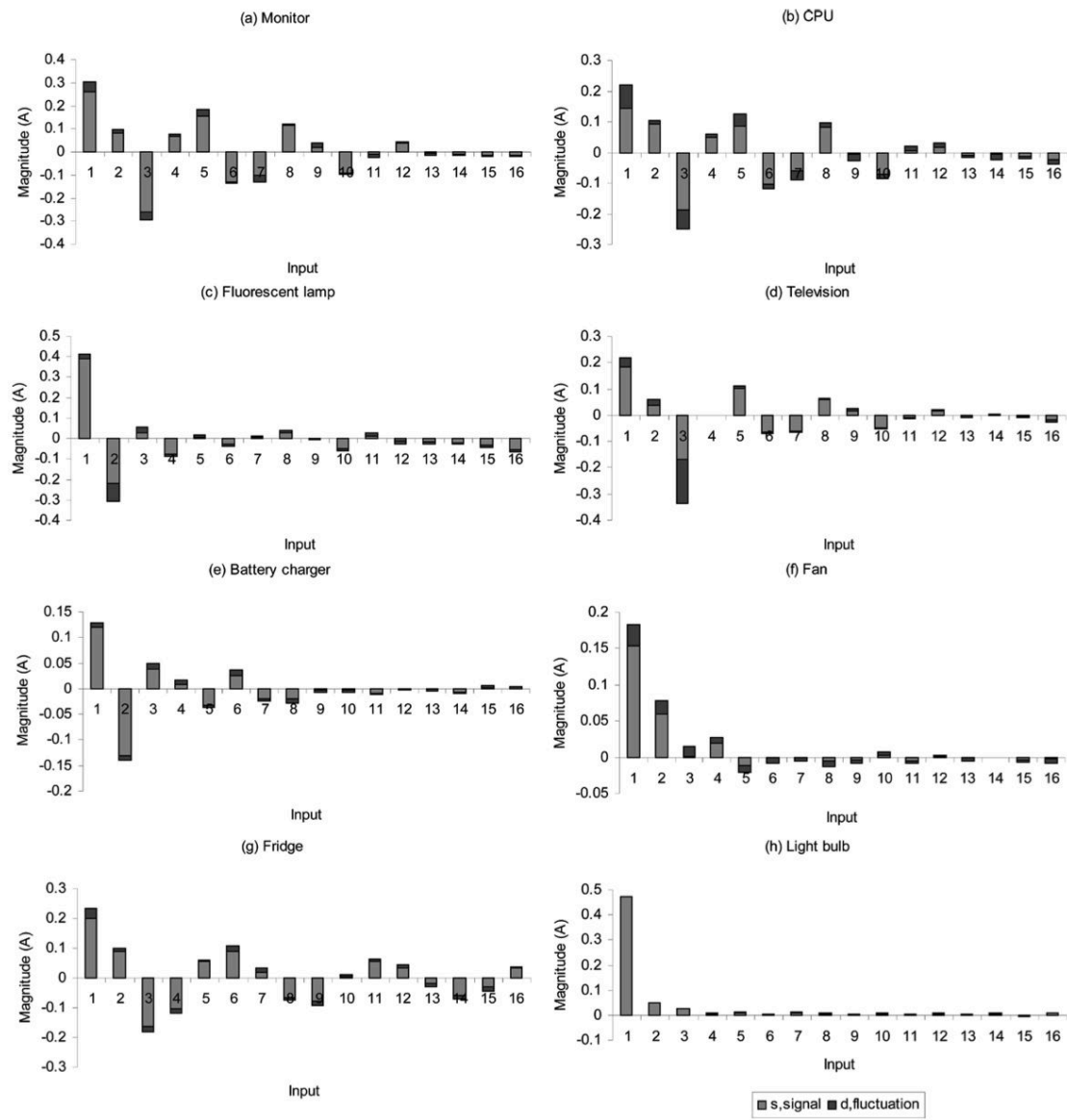


FIGURE 2.14: Harmonic signatures of typical devices [17].

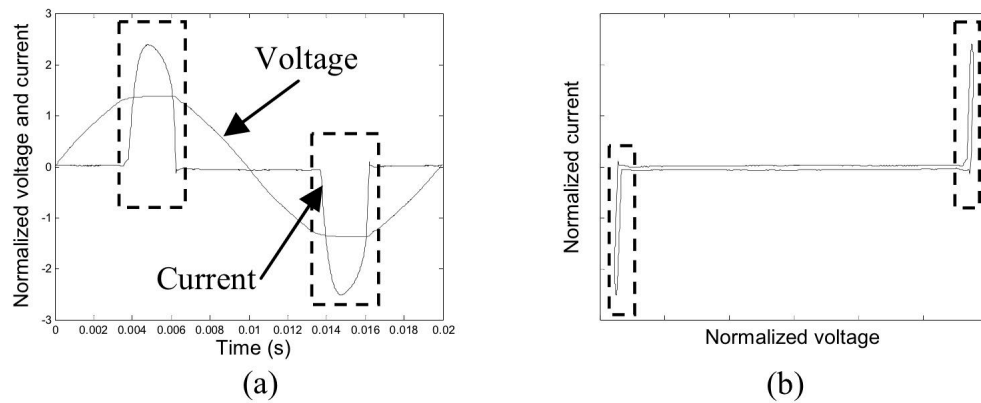


FIGURE 2.15: (a) Current and voltage waveforms of a television. (b) Corresponding V-I trajectory [18].

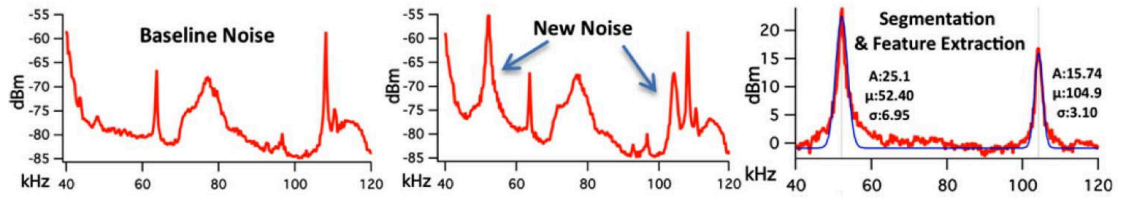


FIGURE 2.16: New noise feature extracted from the background noise [19].

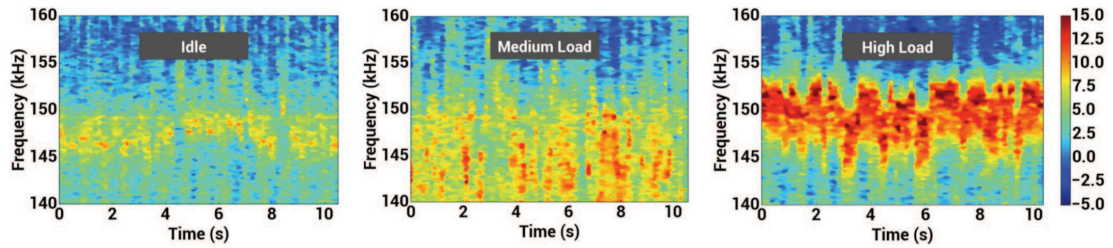


FIGURE 2.17: Time-varying EMI of a laptop at idle, medium load and high load [20].

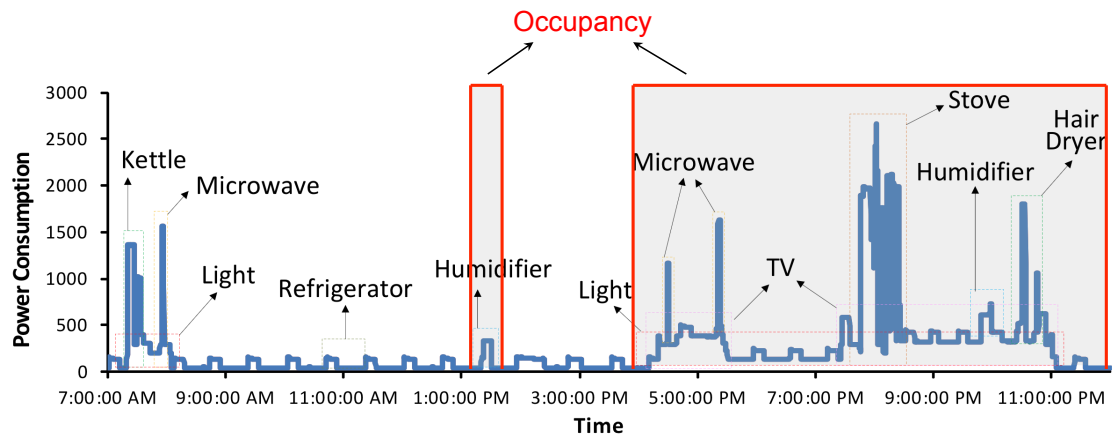


FIGURE 2.18: Power consumption with and without the occupants [21].

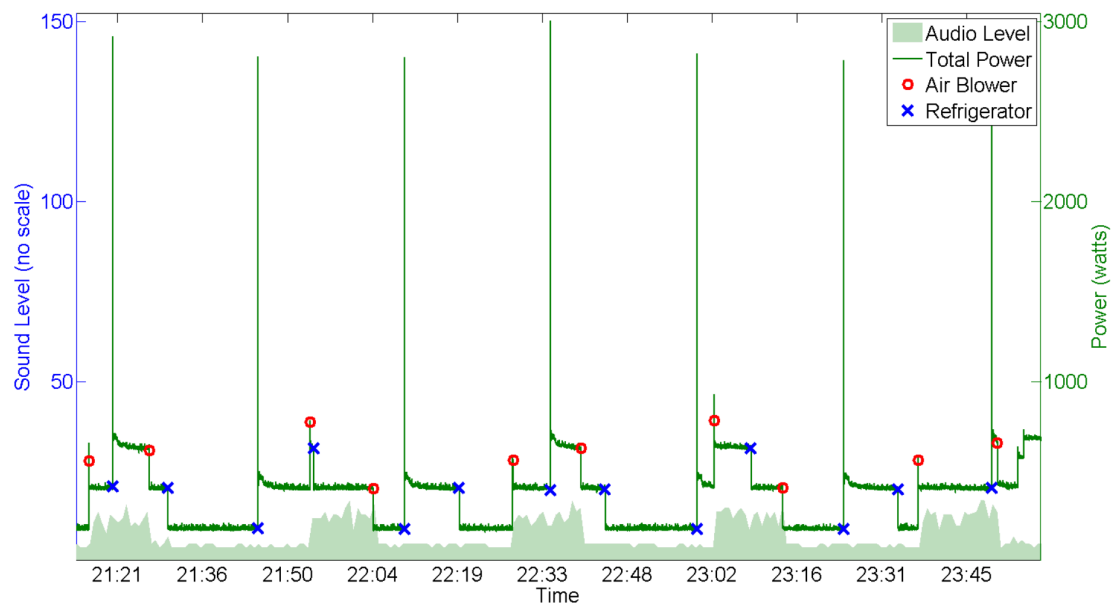


FIGURE 2.19: Aggregate power consumption and the variation of audio signal in a living room [22].

Chapter 3

Algorithms for NILM based on l_1 -norm Minimization

In this chapter, we try to solve the l_1 -norm minimization problem in Eq. (1.2) defined in Section 1.2.2 by brute force approach, also called Least Absolute Error (LAE) based algorithm. Thereby, all possible combinations of states will be sequentially tested and the most suitable one will be selected as solution. However, because of the ambiguity on power demand of different devices, many combinations can give the same or near results, which reduces the algorithm accuracy. To improve performance, we propose to apply an additional information to select the solution among all possible combinations. This is a contribution to research on NILM. Two new methods are studied and introduced in this chapter. The first one, state difference-based algorithm, uses the Hamming distance between the possible combinations and the previous state of devices to find the solution for the l_1 -norm minimization problem, as introduced in Section 3.2. Meanwhile, the second one, called state transition probability-based algorithm and introduced in Section 3.3, considers the state transition probability from previous state to current state to determine the most suitable combination. The three proposed algorithms will be simulated in Matlab with data retrieved from a set of devices in our laboratory. Simulation results will be then compared with edge detection algorithm [1].

3.1 Least Absolute Error based Algorithm

The $l1$ -norm minimization problem for Equation (1.2) is presented as follows:

$$\begin{aligned} \min_{\mathbf{s}} \quad & \| x(t) - \sum_{i=1}^N \sum_{j=1}^{m_i} s_{ij}(t) \times w_{ij} \| \\ \text{subject to:} \quad & s_{ij}(t) \in \{0, 1\}, i = 1, \dots, N, j = 1, \dots, m_i \\ & \sum_{j=1}^{m_i} s_{ij}(t) \in \{0, 1\}, i = 1, \dots, N. \end{aligned} \quad (3.1)$$

To find the solution for problem (3.1), the conditions are at first processed by presenting all possible combinations of state vector \mathbf{s} in matrix \mathbf{S} . Each row of \mathbf{S} corresponds to a combination. For example, if there are two devices and each of them has two operation states, the state matrix \mathbf{S} is defined as

$$\mathbf{S} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

Then, all combinations will be sequentially applied to the following equation to calculate the absolute error between the aggregate power and the total power demand of devices with corresponding state defined as

$$e(k) = |x(t) - \mathbf{s}^{(k)} \times \mathbf{w}^T|, \quad (3.2)$$

where $e(k)$ is the absolute error corresponding to the combination $s^{(k)}$ in row k of matrix \mathbf{S} . The combination giving the least value of $e(k)$ is selected as the solution. Therefore, this method is also called Least Absolute Error (LAE) based algorithm and presented in Algorithm 2. Obviously, the LAE based algorithm is exponentially complex and may be

intractable with a large number of devices. However, it is considered in this chapter to study the performance improvement with the extrinsic information.

Algorithm 2 LAE-based algorithm for $l1$ -norm minimization problem.

```

1: function LAESOLVE( $x, \mathbf{w}$ )
2:   Find all possible combinations of  $\mathbf{s}$  and save in matrix  $\mathbf{S}$ 
3:    $l = \text{length}(x), x(0) = 0$ 
4:   for  $t = 1, \dots, l$  do
5:      $\mathbf{s} = \arg \min_{\mathbf{s}^{(k)} \in \mathbf{S}} |x(t) - \mathbf{s}^{(k)} \times \mathbf{w}^T|$ 
6:      $\mathbf{S}_{\text{out}}(t) = \mathbf{s}$ 
7:   end for
8:   output = matrix  $\mathbf{S}_{\text{out}}$ 
9: end function

```

3.2 State Difference based Algorithm

In real condition, power consumption of a device is not stable. It can vary around an average value due to noise. In order not to leave out the true combination, it is necessary to retain the combinations giving an error close to the least absolute error value and apply an additional constraint to select the solution: among all possible combinations, the final solution gives the smallest Hamming distance with the previous state. Hamming distance between two binary vectors is defined as the number of positions at which the corresponding values are different. A combination at row k of \mathbf{S} is retained if it satisfies the following condition:

$$\frac{|e(k) - e_{LAE}|}{e_{LAE}} \leq \gamma, \quad (3.3)$$

where e_{LAE} denotes the least absolute error and γ is an empirically chosen threshold. If this value is too small, the true solution of state vector \mathbf{s} may be left out, while large value increases the number of retained combinations and the risk of error detection. Therefore, the selection of γ needs to consider the standard deviation of the power consumption of devices as well as the difference between their power demand. Other combinations not satisfying condition (3.3) will be ignored. Finally, the combination having the smallest distance to the previous state is selected as the current state of devices, i.e.,

$$\mathbf{s} = \arg \min_{\mathbf{s} \in \mathbf{S}_{\mathbf{C}}} |\mathbf{s} \oplus \mathbf{s}(t-1)|, \quad (3.4)$$

where $\mathbf{S}_{\mathbf{C}}$ is the set of combinations satisfying condition (3.3). The step-by-step procedure for state difference based method for $l1$ -norm minimization problem is presented in

Algorithm 3. The retention of the combinations giving an absolute error close to the least one can prevent the algorithm from rejecting the accurate solution. However, this also increases the amount of computations to process the retained combinations, depending on the ambiguity level of the set of devices.

Algorithm 3 State difference based algorithm for $l1$ -norm minimization problem.

```

1: function DIFFSOLVE( $x, \mathbf{w}, \gamma$ )
2:   Find all possible combinations of  $\mathbf{s}$  and save in matrix  $\mathbf{S}$ 
3:    $l = \text{length}(x), x(0) = 0, K = \text{length}(\mathbf{S})$ 
4:   for  $t = 1, \dots, l$  do
5:     for  $k = 1, \dots, K$  do
6:        $e(k) = |x(t) - \mathbf{s}^{(k)} \times \mathbf{w}^T|$ 
7:     end for
8:      $e_{LAE} = \min \{e(i) | i = 1, \dots, K\}$ 
9:     Find  $\mathbf{S}_C \subset \mathbf{S} : \forall \mathbf{s}^{(i)} \in \mathbf{S}_C, |e(i) - e_{LAE}| \leq \gamma \times e_{LAE}$ 
10:     $\mathbf{s} = \arg \min_{\mathbf{s}^{(i)} \in \mathbf{S}_C} \{|\mathbf{s}^{(i)} \oplus \mathbf{s}(t-1)|\}$ 
11:     $\mathbf{S}_{\text{out}}(t) = \mathbf{s}$ 
12:  end for
13:  output = matrix  $\mathbf{S}_{\text{out}}$ 
14: end function

```

3.3 State Transition Probability based Algorithm

Different from the previous algorithm, this method considers the state transition probability of each device to determine the current state instead of calculating the Hamming distance from all suitable combinations to the previous state of devices. Transition probability of each device, as shown in Figure 3.1, is obtained from the training period. Based on condition (3.3), a subset of suitable combinations is retained and the coincidence probability for each one is calculated by the product of state transition probability of each element device, i.e.,

$$Pr(\mathbf{s}(t) | \mathbf{s}(t-1)) = \prod_{i=1}^N p((s_i(t) | s_i(t-1))), \quad (3.5)$$

where $p((s_i(t) | s_i(t-1)))$ denotes the state transition probability of device i . The combination giving the largest probability will be selected to determine the current state of devices. Algorithm 4 details the state determination in $l1$ -norm minimization problem based on the state transition probability. Similar to the state difference based algorithm, this method also needs more computations to calculate the coincidence transition probability of the retained combinations.

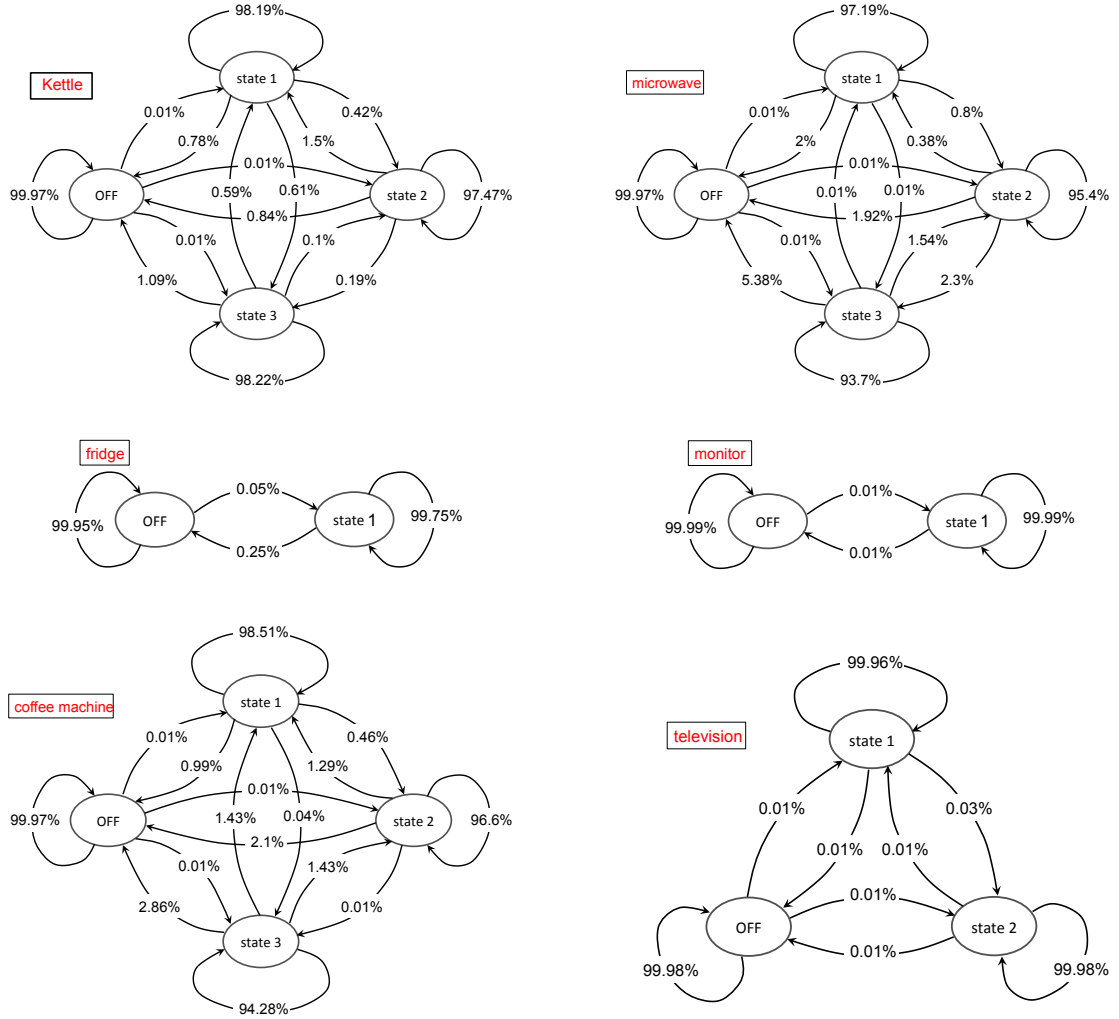


FIGURE 3.1: State transition probability of some devices in Athenium dataset obtained from a training period by attaching each of them to a power meter.

Algorithm 4 State transition probability based algorithm for $l1$ -norm minimization problem.

```

1: function PROBSOLVE( $x, \mathbf{w}, \gamma$ )
2:   Find possible combinations of  $\mathbf{s}$  and save in matrix  $\mathbf{S}$ 
3:    $l = \text{length}(x), x(0) = 0, K = \text{length}(\mathbf{S})$ 
4:   for  $t = 1, \dots, l$  do
5:     for  $k = 1, \dots, K$  do
6:        $e(k) = |x(t) - s^{(k)} \times w^T|$ 
7:     end for
8:      $e_{LAE} = \min \{e(i) | i = 1, \dots, K\}$ 
9:     Find  $\mathbf{S}_C \subset \mathbf{S} : \forall \mathbf{s}^{(i)} \in \mathbf{S}_C, |e(i) - e_{LAE}| \leq \gamma \times e_{LAE}$ 
10:     $\mathbf{s} = \arg \max_{\mathbf{s}^{(i)} \in \mathbf{S}_C} \{Pr(\mathbf{s}^{(i)} | \mathbf{s}(t-1))\}$ 
11:     $\mathbf{S}_{out}(t) = \mathbf{s}$ 
12:   end for
13:   output = matrix  $\mathbf{S}_{out}$ 
14: end function

```

3.4 Experimental Results

In this section, the three proposed algorithms for $l1$ -norm minimization problem will be applied to our *Athemium dataset* retrieved from a monitoring system presented in the next section. The performance of these algorithms will then be compared with the edge detection one [1] to evaluate the improvement. Besides, they are also applied to detect the state of devices in the Reference Energy Disaggregation Dataset (REDD) [4], one of the biggest publicly available dataset for NILM research community. In REDD, the power consumption in two main phases of electricity in six households in the USA is measured every second during many days and the power consumption of some typical devices is retrieved every 3-4 seconds.

3.4.1 Athemium data collection

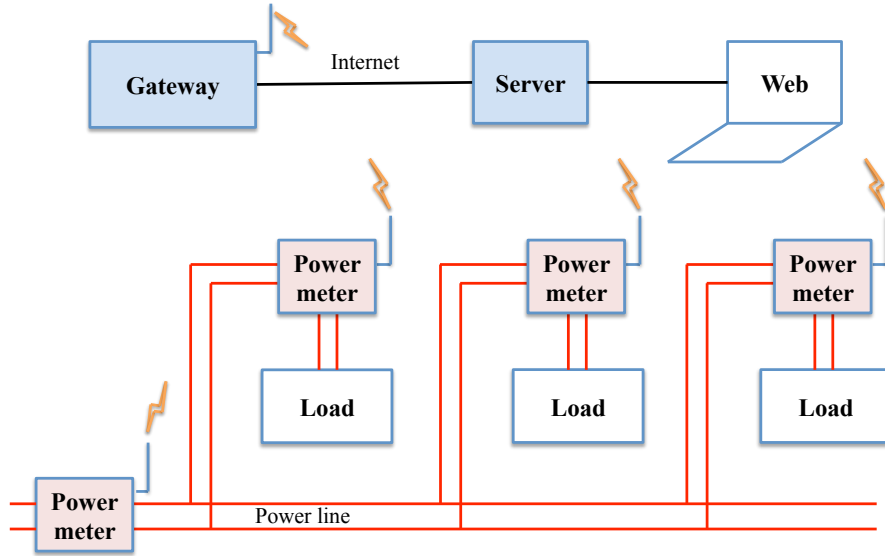
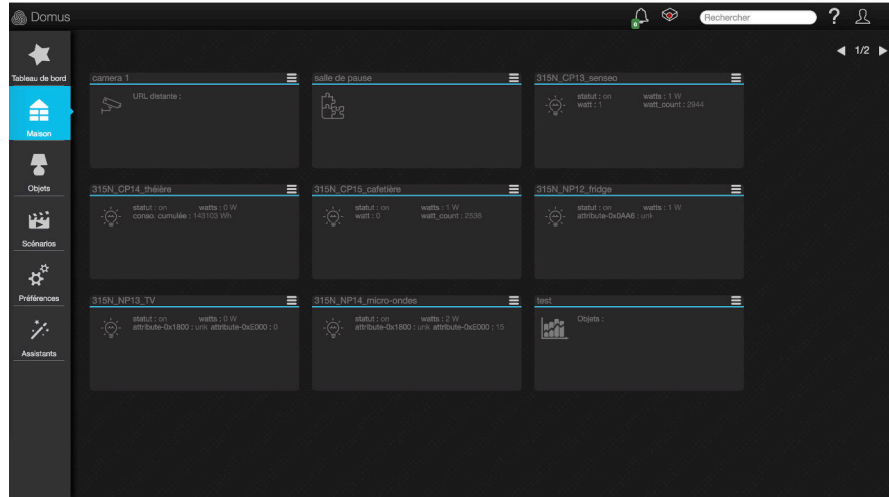


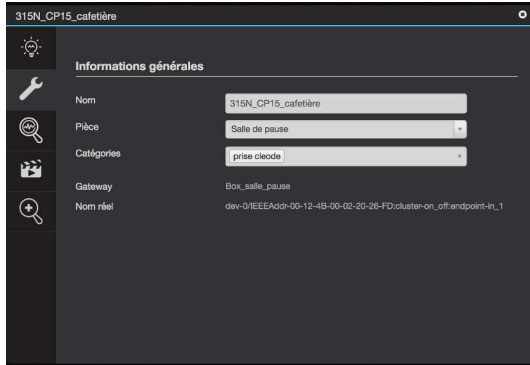
FIGURE 3.2: Smart meters network to retrieve the Athemium data.

To retrieve the Athemium data, a meter network supporting Zigbee wireless communications provided by Athemium company [79] is deployed in the coffee room of our laboratory. The smart meter network is illustrated in Figure 3.2, in which each device is attached to an individual meter. Data provided by these meters are used to learn the characteristics of each device as well as considered as ground truth data to evaluate performance of disaggregation algorithms. Additionally, a global power meter is also installed at the main circuit to measure the aggregate power consumption, which will be used to detect the operating states of each device. These power meters, fabricated

by Netvox Technology Company [80], support wireless communication to send the power measurement to a gateway, which will forward it to an Athemium server through Internet connection. A web interface allows users to observe and download data. This interface lists all deployed power meters and sensors as well as their detailed parameters such as name, position, label, MAC address, as illustrated in Figure 3.3. Besides, users can also display the power consumption of each device during hours, days, weeks, months and years.



(a) List of deployed meters and sensors



(b) Meters configuration



(c) Power measurement display

FIGURE 3.3: Athemium web interface.

In this experiment, six devices including fridge, coffee machine, kettle, microwave, television and screen monitor are connected to power meters, as shown in Figure 3.4. Other loads such as lamps, telephone chargers, Internet modem, outlets, etc., are considered as noise sources. Table 3.1 shows the power demand of each device, while an example of their daily power consumption as well as aggregate power is illustrated in Figure 3.5. Based

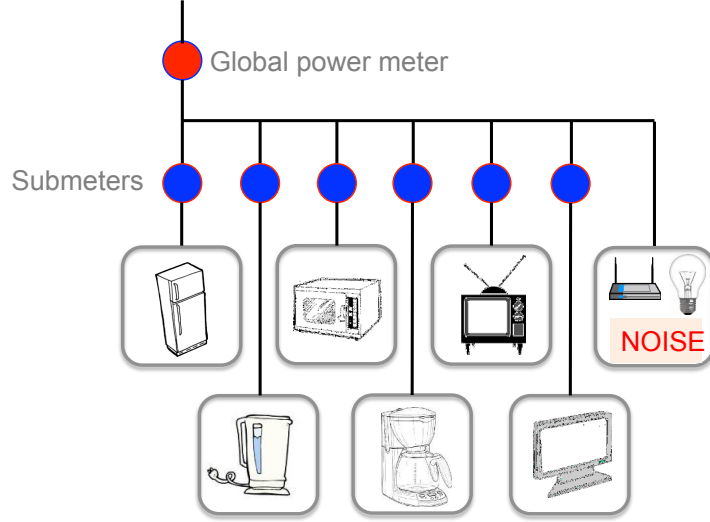


FIGURE 3.4: Athemium power measurement: a global power meter measures the aggregate power consumption and some sub-meters provide training data and ground truth data of each device.

TABLE 3.1: Power demand of devices in Athemium dataset.

| Power demand (Watt) | State 1 | State 2 | State 3 |
|---------------------|---------|---------|---------|
| Fridge | 75 | | |
| Coffee machine | 823 | 797 | 214 |
| Kettle | 1667 | 1692 | 1635 |
| Microwave | 1350 | 1316 | 1290 |
| Television | 203 | 29 | |
| Screen monitor | 72.7 | | |

on these retrieved data, state transition probability of each device can be calculated as presented in Figure 3.1.

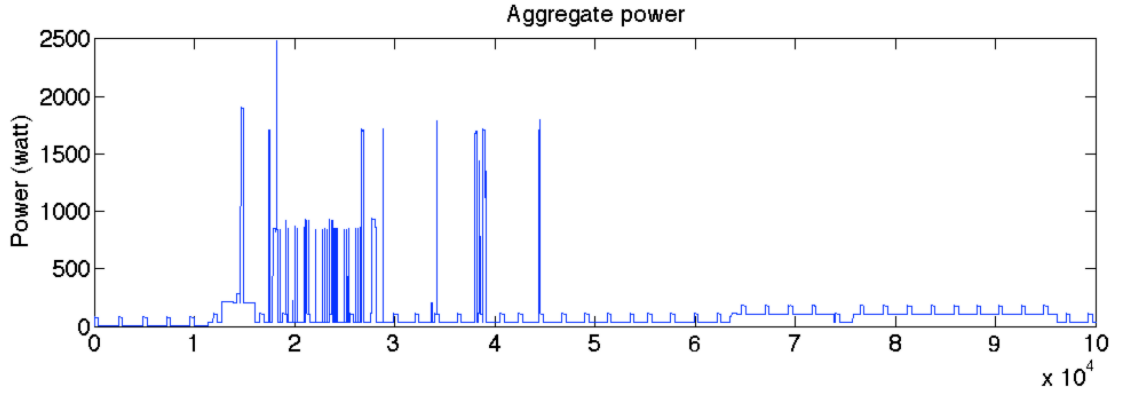
3.4.2 Results and Evaluation

To evaluate the algorithms, three metrics are used including *precision* (pr), *recall* (rc) and *F-measure* (Fm) [81], which are theoretically defined as:

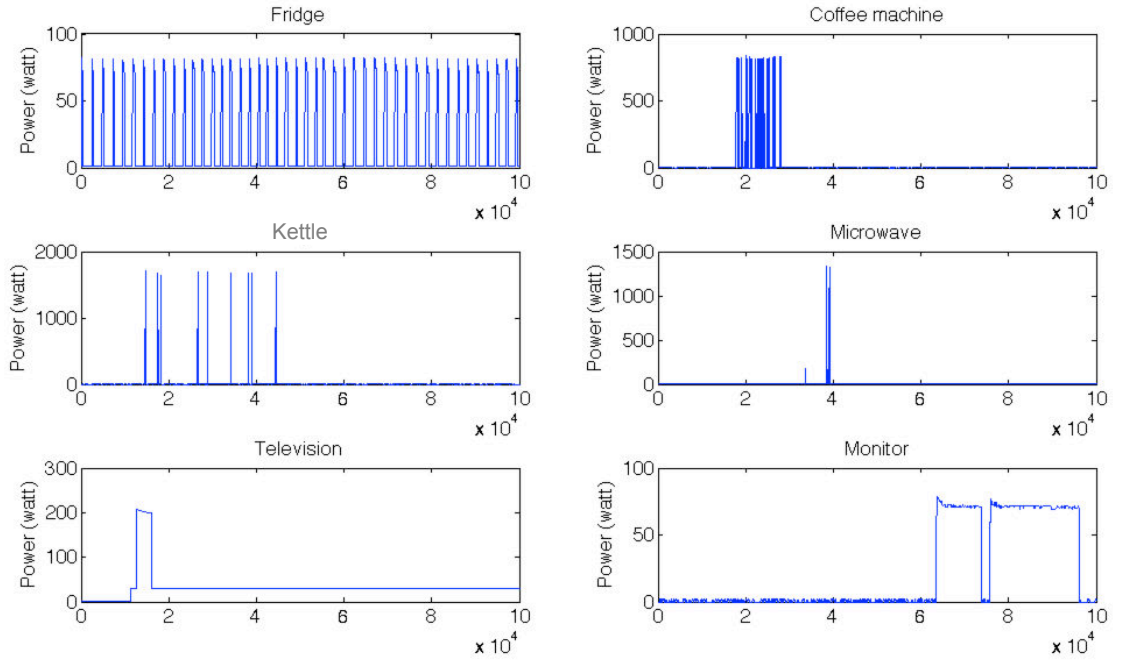
$$pr = \frac{TP}{TP + FP}, \quad (3.6)$$

$$rc = \frac{TP}{TP + FN}, \quad (3.7)$$

$$Fm = \frac{2 \times pr \times rc}{pr + rc}, \quad (3.8)$$



(a) Aggregate power consumption



(b) Power consumption of each individual device

FIGURE 3.5: Power measurement of the devices in Athemium dataset during one day.

where TP , FP and FN are the number of true positives, false positives and false negatives, respectively. A true positive is a true detection of an event, while a false positive (false negative) means a non-event (event) being not correctly detected. As a consequence, the precision can be considered as the reliability of a detected event and the recall is the sensibility to the events of algorithms. Meanwhile, F-measure is interpreted as a weighted average of the precision and recall, which reaches its best value at 1 and worst at 0. For example, if there are 1500 of 2000 time instants that a fridge operates are correctly detected, we have $TP = 1500$ and $FN = 500$. Similarly, $FP = 200$ if 200 instants that the device is turned off are detected as on. As a consequence, we can obtain:

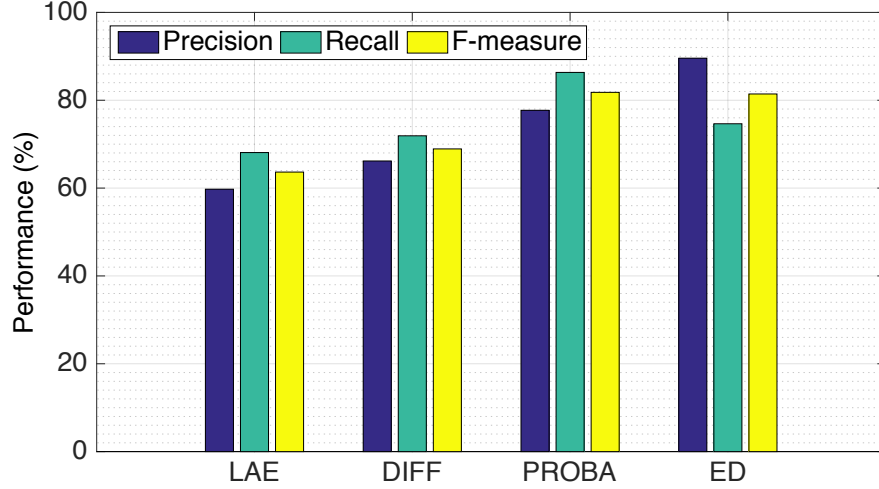


FIGURE 3.6: Performance of the proposed algorithms including LAE, state difference based (DIFF) and transition probability based (PROBA) compared with edge detection algorithm (ED) applied to Athemium data.

$$\begin{aligned}
 pr &= \frac{1500}{1500 + 200} = 0.88, \\
 rc &= \frac{1500}{1500 + 500} = 0.75, \\
 Fm &= \frac{2 \times 0.88 \times 0.75}{0.88 + 0.75} = 0.81.
 \end{aligned}$$

In our Matlab simulation, 30% of data are used to learn the parameters while state detection algorithms are applied to the remaining. In Figure 3.6, performance of the three proposed algorithms with Athemium dataset is shown in comparison with the edge detection algorithm. Apparently, the brute force algorithm without any additional information shows a worse performance than the edge detection approach. While the edge detection algorithm can detect the devices with an overall precision of 89.58% and recall of 74.65%, the LAE based algorithm only shows a poor performance of 60% and 69%, respectively. This result comes from the fact that there are some devices consuming a close power level, e.g. coffee machine (214W) and television (203W). Nevertheless, because the coffee machine has three states, the detection based on the edges height is more effective. This is also the reason why considering the state transition of the devices, executed by selecting a subset of suitable combinations giving an error closed to the least value and applying the difference between each combination with the previous state as a criterion to select the solution, can increase the performance of the brute force algorithm. Concretely, the precision can be improved from 60% to 67% and the recall from 69% to

TABLE 3.2: Detailed performance of the three proposed algorithms including LAE based, state difference based (DIFF) and state transition probability based (PROBA) to detect each device: fridge (FR), coffee machine (CF), kettle (TE), microwave (MW), television (TV) and monitor (MO).

| | Precision (%) | | | Recall (%) | | | F-measure (%) | | |
|----|---------------|-------|-------|------------|-------|-------|---------------|-------|-------|
| | LAE | DIFF | PROBA | LAE | DIFF | PROBA | LAE | DIFF | PROBA |
| FR | 43.08 | 44.24 | 63.80 | 80.88 | 73.95 | 91.86 | 56.22 | 55.36 | 75.30 |
| CM | 76.64 | 80.25 | 76.34 | 99.05 | 98.94 | 98.90 | 86.42 | 86.62 | 86.17 |
| TE | 96.89 | 98.97 | 98.98 | 44.01 | 62.88 | 73.61 | 60.53 | 76.90 | 84.43 |
| MW | 44.30 | 47.01 | 57.63 | 76.22 | 81.55 | 81.55 | 56.03 | 59.64 | 67.53 |
| TV | 78.25 | 86.99 | 83.05 | 95.74 | 95.44 | 95.79 | 86.12 | 91.02 | 89.97 |
| MO | 55.52 | 57.80 | 89.83 | 54.99 | 59.86 | 70.71 | 55.25 | 56.67 | 79.13 |

71%. Especially, the performance improvement is more significant if the state transition probability is applied instead of the Hamming distance between the state combinations. Although performing a lower precision (79% *vs.* 89.58%), the probability based algorithm shows a better recall (87%) than the edge detection one (74.65%), which results in a close F-measure (81.8% *vs.* 81.43%) for both algorithms.

Table 3.2 details the performance of the proposed algorithms correlated to each individual device. Apparently, the LAE based algorithm shows a high precision in detecting the kettle ($pr = 96.89\%$) while it is sensible to the operation of the coffee machine ($rc = 99.05\%$) and television ($rc = 95.74\%$). The fridge and microwave with many spikes on the power signal are identified with a lower precision than other devices ($pr = 43.08\%$ and $pr = 44.30\%$, respectively). Meanwhile, the state difference based algorithm allows to significantly improve the performance of the television (F-measure from 86.12% to 91.02%) and kettle (60.53% to 76.9%). With the transition probability based algorithm, the remarkable increase of the performance corresponds to the fridge (56.22% to 75.30%), kettle (60.53% to 84.43%), microwave (56.03% to 67.53%) and monitor (55.25% to 79.13%).

As mentioned in the previous section, an important parameter affecting the performance of the state difference based and state transition probability based algorithms for $l1$ -norm minimization problem is the threshold γ to select the retained combinations. If this threshold is too small, the true combination may be left out. In contrast, if it is too high, too many combinations are maintained and that may cause the decrease of the performance as well as the increase of the execution time. Figure 3.7 shows the effect

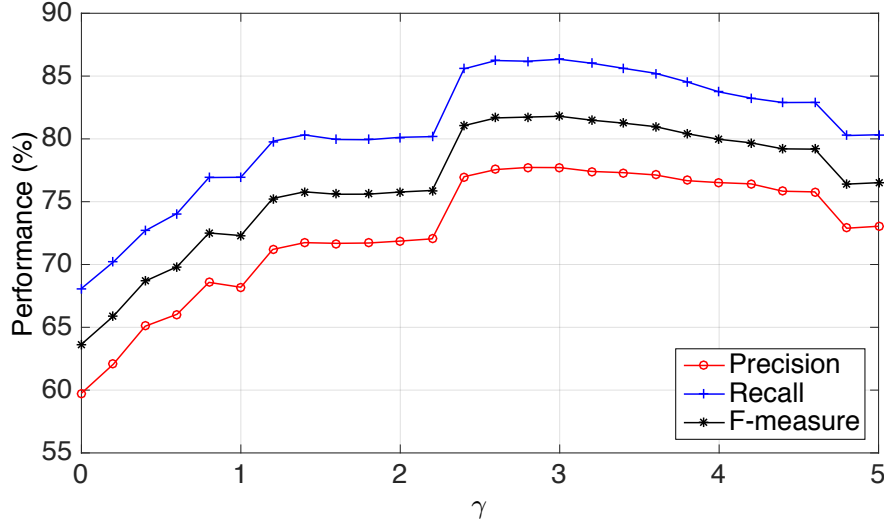


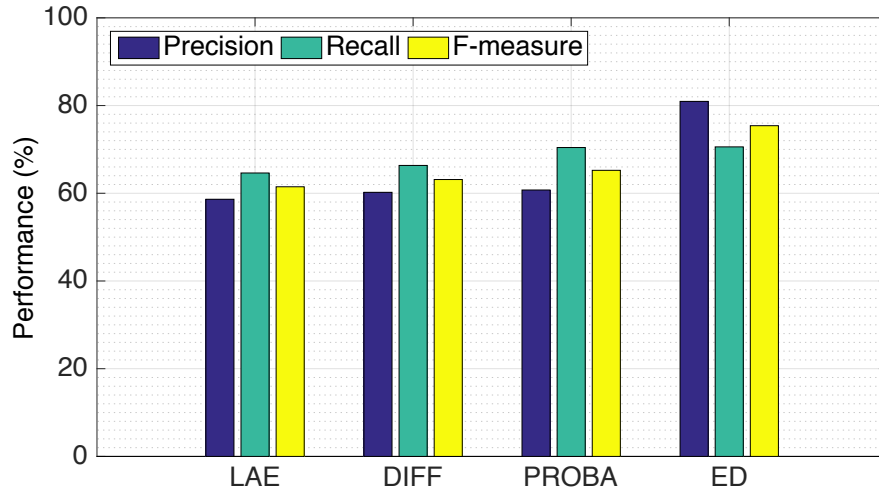
FIGURE 3.7: Effect of threshold γ on the performance of state transition probability based algorithm simulated with Athemium dataset.

of the threshold γ on the performance of the algorithm based on the state transition probability with the best result obtainable at $\gamma = 3$.

Although the additional subroutines significantly improve the performance of LAE based algorithm with Athemium data, it is still less effective when applied to other dataset with more confusions among combinations as well as too much *ghost power* consumed by unmonitored devices such as REDD House 1 [4], as illustrated in Figure 3.8. In this dataset, besides the devices connected to an individual power meter, the total power consumption of other unmonitored devices can be considered as consumed by a unique unknown device. Though there are over 20 channels, we use only nine channels as shown in Table 3.3, because the others have no activation of devices. Besides, two main phases of electricity are combined to have a unique main power signal. As shown in Figure 3.8, the precision of $l1$ -norm minimization based algorithm can be improved from 58.63% to 60.75% and the recall from 64.62% to 70.43% with state transition probability. Nevertheless, they are still lower than the performance given by the edge detection one, with $pr = 80.95\%$ and $rc = 70.58\%$, respectively. Therefore, in the next chapter, a new monitoring system will be used to significantly improve the performance of NILM algorithms based on probability information of each device provided by an additional sensor network.

TABLE 3.3: Power demand of devices in REDD dataset *House 1*.

| Power demand (watt) | State 1 | State 2 | State 3 |
|---------------------|---------|---------|---------|
| Oven | 4140 | 4075 | 3377 |
| Fridge | 193 | 423 | |
| Dish washer | 214 | 1107 | |
| Lighting | 169 | 275 | 345 |
| Microwave | 1531 | 1571 | 1396 |
| gfi bathroom | 1595 | | |
| Outlet | 1056 | | |
| Outlet | 1521 | | |
| Washer dryer | 5097 | 5250 | 5008 |

FIGURE 3.8: Performance of the proposed algorithms including LAE, state difference based (DIFF) and transition probability based (PROBA) compared with the edge detection (ED) algorithm applied to REDD *House 1*.

3.5 Conclusions

In this chapter, we directly solve the $l1$ -norm minimization problem in NILM by applying three proposed algorithms, including:

- LAE-based algorithm: this algorithm is a brute force method to apply all possible combinations of states to calculate the absolute error between the aggregate power and the total power demand corresponding to each combination. The combination giving the least value of error is selected to determine the state of devices;

- State difference based algorithm: instead of selecting the least absolute error, this algorithm finds all possible combinations giving an error around it and then compares with the previous state to select one having the smallest Hamming distance;
- State transition probability based algorithm: the solution is selected among suitable combinations by considering the state transition probability from the previous state to the current state of each device.

The experimental results show that by applying the state difference and state transition probability to select suitable combination of state, performance of the brute force algorithm is improved and outperforms the edge detection one. Nevertheless, these proposed algorithms show an exponential complexity and may be intractable with larger number of devices. In addition, an excessive training data is necessary to learn the characteristics of each device such as power demand and state transition probability.

Chapter 4

SmartSense: Sensor-Aided Non-Intrusive Load Monitoring

4.1 SmartSense Principle and Model

In NILM, the disaggregation algorithms always try to extract the features from the overall electrical load signal to identify the devices. However, how can we discern the devices with the same power characteristics? For example [15], if the algorithms use the features related to the average power consumption such as power level, length of steady state, step-change, etc., to discriminate the devices, it can lead to a false detection when separating an incandescent light bulb and a desktop computer, which consume the same power of 150W.

To overcome this restriction and improve the performance of the detection algorithms, we propose to use the operating probability of each device as an additional feature. A low-cost and low-power WSN is proposed to be deployed in homes and buildings to monitor the operation of some specific devices. A system combining NILM with such a WSN, called SmartSense, is shown in Figure 4.1. However, different from ViridiScope [6] which is fully intrusive, our approach only monitors a subset of all devices, which makes it less intrusive. On the other hand, the purpose of the WSN is not to directly provide the state of devices but to estimate their state probability to improve the performance of NILM algorithms. Depending on the type of monitored devices, the corresponding sensors are deployed. For example, the operation of light-emitting devices such as screens,

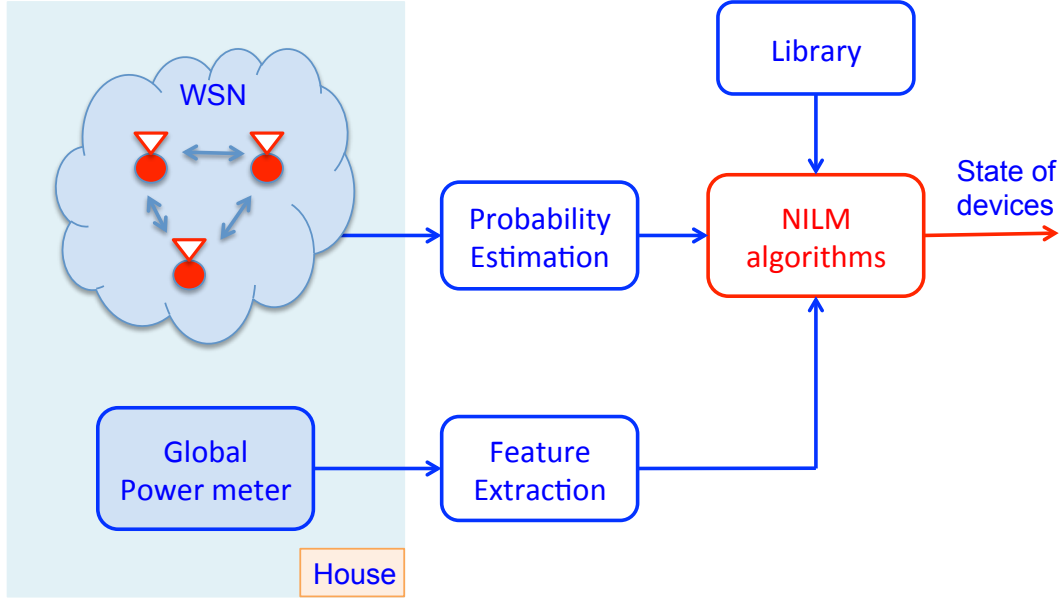


FIGURE 4.1: SmartSense system model: an NILM system combining with a sensor network providing the operating probability of some specific devices.

televisions, lamps can be detected by using the light intensity sensors, while the motor-based devices such as washing machine, fridge can be identified with a vibration sensor. Besides, other types of sensors can also be used to monitor other electrical equipment such as microphones, magnetometers, etc. A local algorithm in the WSN is then constructed to transform the detection of sensors to the probability of the corresponding devices. The probability estimation is executed based on the performance evaluation metrics including *precision* and *negative predictive value*, calculated by

$$pr = \frac{TP}{TP + FP} \quad (4.1)$$

$$npv = \frac{TN}{TN + FN}, \quad (4.2)$$

where pr , TP , FP and FN are defined in Chapter 3 on page 42, while npv and TN denote the negative predictive value and the number of true negatives, respectively. A true negative is defined as a true detection of non-event. Therefore, the negative predictive value can be considered as the reliability of a detected non-event. As a consequence, when a device is determined as running, the probability of on-state and off-state are equal to pr and $(1 - pr)$, respectively. In contrast, when not running, they correspond to $(1 - npv)$ and npv . For other unmonitored devices, all states can occur with the same probability.

To prove the efficiency of the operating probability in improving the performance of

NILM, in this research, we propose two main approaches for the state detection. In the first approach, the minimization problem (3.1) of NILM can be modeled as a Knapsack problem and solved by two proposed algorithms including Compositional Pareto-algebraic Heuristic (CPH) and Dynamic Programming (DP). Meanwhile, the second one applies the probability to two existing algorithms including Edge Detection (ED) [1] and Dynamic Time Warping (DTW) [2]. In the rest of this chapter, the word *event* is used to denote an on-state in the CPH, DP and DTW algorithms and an edge in the ED one.

4.2 Approach 1: Formulation as a Knapsack problem

Let consider again the l_1 -norm minimization problem (3.1) of NILM:

$$\begin{aligned} \min_{\mathbf{s}} E(\mathbf{s}) &= \left| \sum_{i=1}^N \sum_{j=1}^{m_i} w_{ij} s_{ij} - x \right| \\ \text{subject to } s_{ij} &\in \{0, 1\}, i = 1, \dots, N, j = 1, \dots, m_i \\ \sum_{j=1}^{m_i} s_{ij} &\in \{0, 1\}, i = 1, \dots, N, \end{aligned} \quad (4.3)$$

where $E(\mathbf{s})$ can be interpreted as the absolute error between the measured aggregate power consumption and the total power demand of all operating devices. Solving Eq. (4.3) is equivalent to find a vector \mathbf{s} comprising all state indicators s_{ij} , $i = 1, \dots, N, j = 1, \dots, m_i$ of all devices, i.e., $\mathbf{s} = \{s_{11}, \dots, s_{1m_1}, s_{21}, \dots, s_{2m_2}, \dots, s_{N1}, \dots, s_{Nm_N}\}$. In Chapter 3, this problem is solved by applying three proposed methods including least absolute error, state difference and state transition probability. However, in the SmartSense system, by using an additional parameter related to the operating probability, a new problem is formulated from (4.3) and two methods are proposed to solve it including CPH and DP. The on/off state probability of each device is estimated as mentioned in Section 4.1. Nevertheless, as the sensors can only detect if a device is on or off but cannot distinguish its different power states, the operating probability will then be equally divided to all states, i.e. $p_{ij} = p_i/m_i, j = 1, \dots, m_i$. Let consider the problem formulation in the case that each device has only two states, on or off, and in a general case in which each device has a finish number of power states.

4.2.1 Problem formulation

One-state devices Firstly, let consider the simplest case in which each device can only operate at a unique stable state with a corresponding power demand w_i . Denote s_i as the state indicator of device i and notation ⁽¹⁾ for one-state using case, problem (4.3) becomes:

$$\begin{aligned} \min_{\mathbf{s}} E^1(\mathbf{s}) &= \left| \sum_{i=1}^N w_i s_i - x \right| \\ \text{subject to} \quad s_i &\in \{0, 1\}, i = 1, \dots, N, \end{aligned} \quad (4.4)$$

where $\mathbf{s} = \{s_1, s_2, \dots, s_N\}$. Assuming device i is in on-state with probability of $p_i, 0 \leq p_i \leq 1$, the probability mass function is then:

$$Pr(s_i) = p_i^{s_i} (1 - p_i)^{(1-s_i)}. \quad (4.5)$$

Because the operation of N devices is assumed to be independent, the coincidence probability, the probability that all events happen simultaneously, is as follows:

$$\begin{aligned} Pr(\mathbf{s}) &= \prod_{i=1}^N Pr(s_i) \\ &= \prod_{i=1}^N p_i^{s_i} (1 - p_i)^{(1-s_i)}. \end{aligned} \quad (4.6)$$

To reduce the computational complexity, the log-linear form can be applied to transform the multiplication in (4.6) to an addition, which gives:

$$\begin{aligned} L^1(\mathbf{s}) &= -\log(Pr(\mathbf{s})) \\ &= -\log\left(\prod_{i=1}^N p_i^{s_i} (1 - p_i)^{(1-s_i)}\right) \\ &= -\sum_{i=1}^N (s_i \log p_i + (1 - s_i) \log (1 - p_i)). \end{aligned} \quad (4.7)$$

Denote $L^1(s_i) = (1 - s_i)l_{i0}^1 + s_i l_{i1}^1$ with $l_{i0}^1 = -\log(1 - p_i)$ and $l_{i1}^1 = -\log p_i$, Eq. (4.7) can be written as

$$L^1(\mathbf{s}) = \sum_{i=1}^N L^1(s_i). \quad (4.8)$$

By this transformation, the increase of $Pr(\mathbf{s})$ corresponds to the decrease of $L^1(\mathbf{s})$. The problem is now to find a combination of operating devices not only giving the least absolute error $E^1(\mathbf{s})$ but also having the maximum probability, i.e. the minimum value of $L^1(\mathbf{s})$. Therefore, the problem (4.4) is developed to a co-optimization problem as follows:

$$\begin{aligned} & \min_{\mathbf{s}} [P^1(\mathbf{s}) = E^1(\mathbf{s}) + \lambda \times L^1(\mathbf{s})] \\ \text{subject to } & \sum_{i=1}^N w_i s_i \leq x + \epsilon \\ & s_i \in \{0, 1\}, i = 1, \dots, N, \end{aligned} \quad (4.9)$$

where λ is a regularization parameter and empirically chosen, ϵ corresponds to the standard deviation of the aggregate power consumption. The problem (4.9) is a kind of **0-1 Knapsack problem**, in which the aim is to fill the knapsack by selecting among various objects, each of which having a particular weight and giving a particular profit [82]. The optimization problem is then to choose the objects in order to obtain the maximum profit while respecting the knapsack capacity.

Multi-state devices Return to the general case where each device has a finite number of states, the following condition needs to be satisfied:

$$\sum_{j=1}^{m_i} s_{ij} \leq 1, s_{ij} \in \{0, 1\}, i = 1, \dots, N, \quad (4.10)$$

where m_i is the number of states of device i . Therefore, the operating probability of device i , denoted as $Pr(\mathbf{s}_i)$, $\mathbf{s}_i = \{s_{i1}, s_{i2}, \dots, s_{im_i}\}$, is

$$Pr(\mathbf{s}_i) = p_{i0}^{\prod_{j=1}^{m_i} (1-s_{ij})} \times \prod_{k=1}^{m_i} p_{ik}^{\prod_{j=1, j \neq k}^{m_i} s_{ik}(1-s_{ij})}, \quad (4.11)$$

where $p_{i0} = p(s_{ij} = 0, j = 1, \dots, m_i)$ is the probability of off-state of device i and $p_{ik} = p(s_{ik} = 1, s_{ij} = 0, \forall j \neq k)$ is the probability of operating at state k . Intuitively, we have

$\sum_{j=0}^{m_i} p_{ij} = 1$. Transforming Eq. (4.11) to log-linear form gives

$$\begin{aligned} L^M(\mathbf{s}_i) &= -\log Pr(\mathbf{s}_i) \\ &= -\prod_{j=1}^{m_i} (1 - s_{ij}) \log p_{i0} \\ &\quad - \sum_{k=1}^{m_i} \left\{ \prod_{j=1, j \neq k}^{m_i} s_{ik} (1 - s_{ij}) \log p_{ik} \right\}. \end{aligned} \quad (4.12)$$

Denoting $l_{ik}^M = -\log p_{ik}, k = 0, \dots, m_i$, Eq. (4.12) can be rewritten as

$$L^M(\mathbf{s}_i) = \prod_{j=1}^{m_i} (1 - s_{ij}) l_{i0}^M + \sum_{k=1}^{m_i} \left\{ \prod_{j=1, j \neq k}^{m_i} s_{ik} (1 - s_{ij}) l_{ik}^M \right\}. \quad (4.13)$$

Similarly to the case of one-state devices, the concidence probability $Pr(\mathbf{s})$, $\mathbf{s} = \{\mathbf{s}_1, \dots, \mathbf{s}_N\}$, and its respective log-linear form $L^M(\mathbf{s})$ can be formulated as

$$Pr(\mathbf{s}) = \prod_{i=1}^N Pr(\mathbf{s}_i) \quad (4.14)$$

$$L^M(\mathbf{s}) = \sum_{i=1}^N L^M(\mathbf{s}_i). \quad (4.15)$$

As a consequence, the co-optimization problem to minimize the least absolute error in problem (4.3) and maximize the coincidence probability in Eq. (4.14) is modified to

$$\begin{aligned} &\min_{\mathbf{s}} [P^M(\mathbf{s}) = E^M(\mathbf{s}) + \lambda \times L^M(\mathbf{s})] \\ &\text{subject to} \quad \sum_{i=1}^N \sum_{j=1}^{m_i} w_{ij} s_{ij} \leq x + \epsilon \\ &\quad \sum_{j=1}^{m_i} s_{ij} \leq 1, j = 1, \dots, N \\ &\quad s_{ij} \in \{0, 1\}, i = 1, \dots, N, j = 1, \dots, m_i. \end{aligned} \quad (4.16)$$

Considering the state of devices as objects and the aggregate power consumption as the knapsack capacity, Eq. (4.14) is equivalent to the Knapsack problem. The state detection is apparently similar to dividing the available objects into groups and selecting a maximum of one object from each group to fill the knapsack. Hence, the problem (4.16) can be considered as a so-called **Multiple-Choice Knapsack (MCK)** problem [83].

In SmartSense, at each data point, an algorithm will be applied to solve the co-optimization

problem to find the corresponding state of each device. To solve the Knapsack problem, the naive and straightforward approach is brute force, as introduced in Chapter 3. However, the exponential complexity makes it intractable with large number of devices. Besides, there are several other approaches such as branch and bound [84–86], genetic algorithm [87–90], dynamic programming (DP) [83, 91], compositional Pareto-algebraic heuristic (CPH) [3, 92–97], or combining DP with branch and bound [98].

In this study, we focus on CPH and DP, two most effective approaches in Knapsack problem, to apply in the context of SmartSense. These algorithms have some driving parameters allowing to reduce the computational complexity.

4.2.2 Algorithms to solve the SmartSense problem

4.2.2.1 Compositional Pareto-Algebraic Heuristic (CPH)

The CPH algorithm is selected to solve the knapsack problem because it provides a fast and scalable method to solve combinatorial optimization problems. It is based on Pareto algebra principles, where we want to determine which set of solutions is better than the rest. CPH computes solutions in incremental fashion while discarding partial solutions which will not lead to optimal solutions.

CPH is based on a recursive relation and illustrated by an example in Figure 4.2. Figure 4.2(a) presents a set of three devices with different operation states. Each state consumes a stable power with a corresponding probability. Assuming at time instant t , the aggregate power consumption x is equal to 150 W. With the standard deviation ϵ of 5 W, the total power demand of any set of operating devices cannot exceed the bound $R = x + \epsilon = 155$ W. Figure 4.2(b) shows how the CPH algorithm operates. In the first step, each device is represented as a set of *tuples*, each corresponding to a state. For example, device D_1 comprises three tuples: $(0, 23.03, 173.03)$, $(50, 2.23, 102.23)$, $(150, 23.03, 23.03)$. The first element d_1 of each tuple represents the power demand, the second one d_2 contains the product of the operating probability in log-linear form and the regularization parameter λ , equal to 10 in this example, and the third one d_3 is calculated from two

first elements d_1 and d_2 as follows:

$$d_1 = w \quad (4.17)$$

$$d_2 = -\lambda \times \log p \quad (4.18)$$

$$d_3 = |x - d_1| + d_2. \quad (4.19)$$

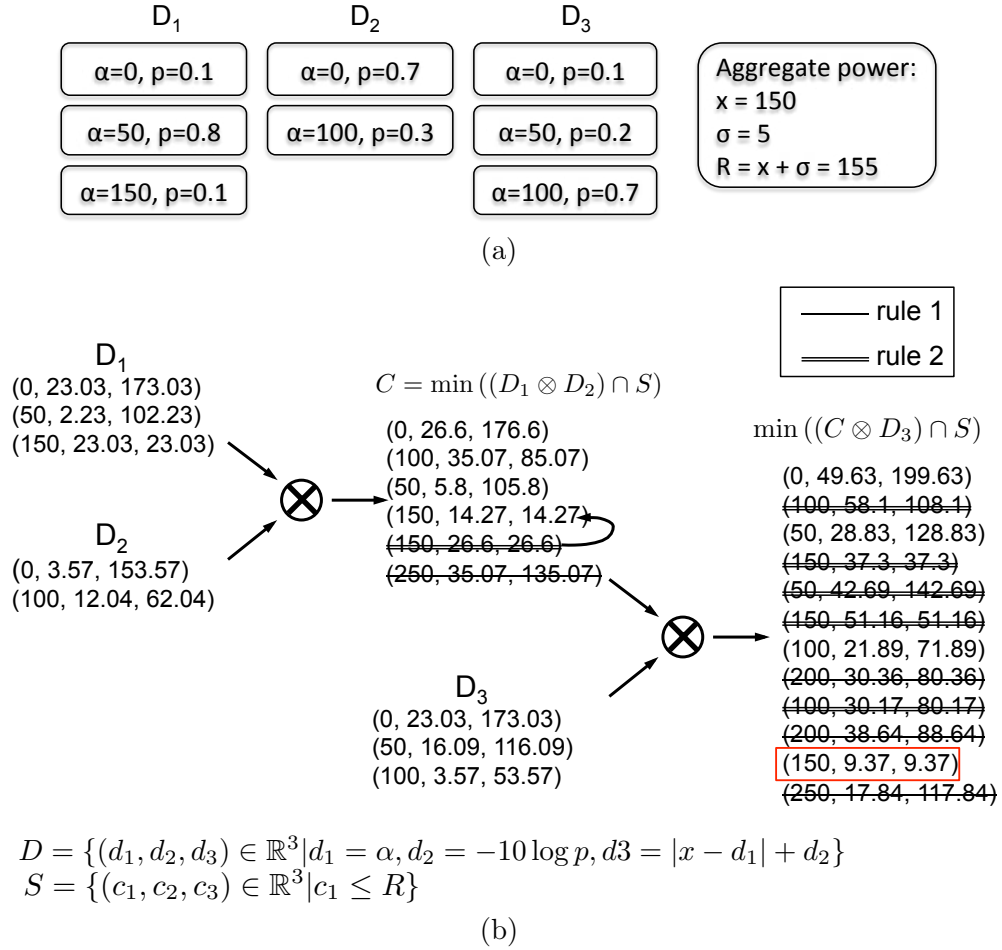


FIGURE 4.2: Running example of CPH. (a) Three devices with corresponding power demand and probability of each state as well as the aggregate power consumption. (b) Each device is represented as a set of tuples, each tuple corresponds to a state. At each iteration, a partial solution can be discarded if its first element (power consumption) exceeds the bound R (rule 1), or its third element is greater than another solution while consuming larger or same power (rule 2). After all sets are combined, the tuple giving the smallest value of the third element is selected as the final solution.

In the second step, the sets of tuples are pairwise and iteratively combined by a so-called *product-sum* operation \otimes , which takes all possible combinations from two sets, sums their first and second elements per dimension and then calculates the third element

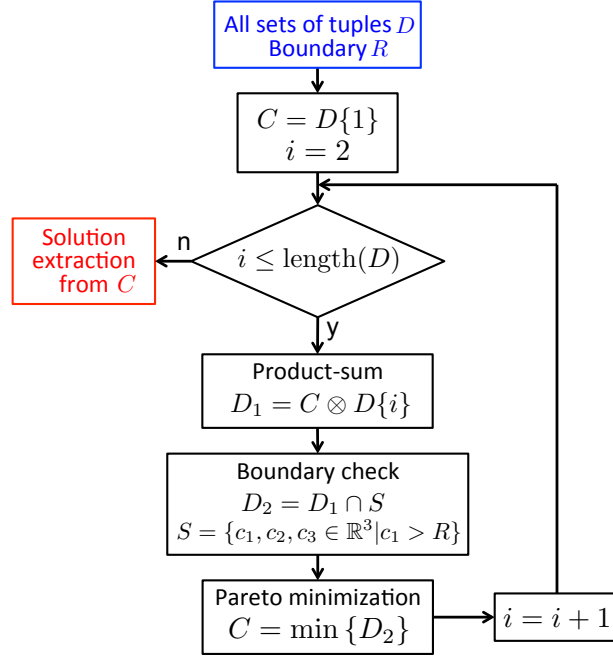


FIGURE 4.3: CPH algorithm flowchart. All sets of tuples are saved in cell D . After the last iteration, the solution is extracted by selecting a tuple in C with the least value of the third dimension.

by Eq. (4.19). Some of the partial solutions will be discarded in the third step by two criteria. The first criterion to reject a tuple is that its power demand violates the bound R (rule 1), denoted by the operation $(D_1 \otimes D_2) \cap S$, and the second one is that this tuple is dominated, i.e. it has a greater third value but the same or larger first value than another tuple (rule 2). This process is called *Pareto minimization* [3] and denoted by the operation $C = \min((D_1 \otimes D_2) \cap S)$, where C contains the remaining solutions called *Pareto points*. In the final step, after all sets are combined and the Pareto points are found, the point giving the smallest value in the third element is selected as the final solution. The CPH algorithm always finds all feasible solutions after the final step. Therefore, the results are not affected by the order of load selection at each step.

Figure 4.3 resumes the operation of CPH algorithm with the procedure to combine two sets of tuples presented in Algorithm 5 and the Pareto minimization illustrated in Algorithm 6. The notations $D_1 \leq D_2$ and $D_1 < D_2$ mean that the tuple D_1 is dominated and strictly dominated, respectively, by D_2 .

Algorithm 5 Combine two sets of tuples.

```

1: function ALLCOMB( $D_1, D_2, x$ )
2:    $l1 = \text{length}(D_1)$ 
3:    $l2 = \text{length}(D_2)$ 
4:   for  $i = 1, \dots, l1$  do
5:     for  $j = 1, \dots, l2$  do
6:        $D\{(i-1)l_1 + j\}(1) = D_1\{i\}(1) + D_2\{j\}(1)$ 
7:        $D\{(i-1)l_1 + j\}(2) = D_1\{i\}(2) + D_2\{j\}(2)$ 
8:        $D\{(i-1)l_1 + j\}(3) = |D\{(i-1)l_1 + j\}(1) - x| + D\{(i-1)l_1 + j\}(2)$ 
9:     end for
10:  end for
11:  output =  $D$ 
12: end function

```

Algorithm 6 Pareto minimization.

```

1: function PARETOMIN( $D, R$ )
2:    $l = \text{length}(D)$ 
3:    $reject = \text{zeros}(l, 1)$ 
4:   for  $i = 1, \dots, l$  do
5:     if  $D\{i\}(1) > R$  then
6:        $reject(i) = 1$ 
7:     end if
8:   end for
9:    $Ind = \text{find}(reject \neq 1)$ 
10:   $D1 = D\{Ind\}$ 
11:   $l1 = \text{length}(D1)$ 
12:   $reject1 = \text{zeros}(l1, 1)$ 
13:  for  $i = 1, \dots, l1 - 1$  do
14:    for  $j = i + 1, \dots, l1$  do
15:      if  $reject(j) \neq 1$  and  $D1\{i\} < D1\{j\}$  then
16:         $reject(i) = 1$ 
17:      else if  $reject(j) \neq 1$  and  $D1\{j\} \leq D1\{i\}$  then
18:         $reject(j) = 1$ 
19:      end if
20:    end for
21:  end for
22:   $Ind1 = \text{find}(reject1 \neq 1)$ 
23:   $D2 = D1\{Ind1\}$ 
24:  output =  $D2$ 
25: end function

```

4.2.2.2 Dynamic Programming (DP)

The principle of the DP algorithm to solve the Knapsack problem is that, if the last item is rejected, the best profit will only depend on the remaining items with the full capacity of the knapsack. In contrast, if it is selected, the best profit includes the profit of this item and the best profit obtainable from the remaining items with the remaining capacity of knapsack after subtracting the weight of the last item.

Therefore, the DP procedure is to construct a profit table with each row corresponding to an item and the number of columns depending on the knapsack capacity with unit step. The profit value of each cell will be calculated based on the current capacity, the corresponding item as well as the recursive relation with the previous cells, which can be explained as follows. At any cell corresponding to an item and a value of capacity, if the item is heavier than the available capacity, it will be rejected and the profit value of the cell is obtained from the previous items. In contrast, it is necessary to compare the profit obtainable with and without that item. The set of the previous items must ensure enough capacity if new item is selected.

One-state devices In the context of one-state devices, each device is considered as an item with its power demand as weight, while the profit is computed from the power demand as well as the operating probability. Although the absolute error between the aggregate power and the total power demand of the identified devices is a criterion to identify the operating devices, it does not satisfy the recursive relation. Concretely,

$$\left| \sum_{j=1}^i w_j s_j - x \right| \neq w_i s_i + \left| \sum_{j=1}^{i-1} w_j s_j - x \right| \quad (4.20)$$

if $w_i s_i \times \left(\sum_{j=1}^{i-1} w_j s_j - x \right) < 0$.

To overcome this restriction, we construct two separated tables, P_e^1 and P_l^1 , for the power and the probability parameters, respectively. Each table is composed of N rows and R columns with $R = \lceil x + \epsilon \rceil$, where $\lceil \cdot \rceil$ is the rounding up operator. The entry of the first table $P_e^1(i, \beta) = \sum_{j=1}^i w_j s_j - x$ and second table $P_l^1(i, \beta) = \lambda \times \sum_{j=1}^i L^1(s_j)$ are calculated by Algorithm 7. A flag table F is also constructed to mark if a device is running or not. In Algorithm 7, because the weight of an item is not an integer number, the notation $\lceil \cdot \rceil$

Algorithm 7 Entry calculation for profit tables in case of one-state devices.

Ensure: $P_e^1(i, \beta), P_l^1(i, \beta)$

- 1: $a_1^1 = P_e^1(i-1, \lfloor \beta - w_i \rfloor) + w_i$
- 2: $a_2^1 = P_l^1(i-1, \lfloor \beta - w_i \rfloor) + \lambda \times l_{i1}^1$
- 3: $b_1^1 = P_e^1(i-1, \beta)$
- 4: $b_2^1 = P_l^1(i-1, \beta) + \lambda \times l_{i0}^1$
- 5: $A^1 = \lfloor a_1^1 \rfloor + a_2^1$
- 6: $B^1 = \lfloor b_1^1 \rfloor + b_2^1$
- 7: $F(i, \beta) = 0$
- 8: **if** $w_i \leq \beta$ and $A^1 \leq B^1$ **then**
- 9: $P_e^1(i, \beta) = a_1^1$
- 10: $P_l^1(i, \beta) = a_2^1$
- 11: $F(i, \beta) = 1$
- 12: **else**
- 13: $P_e^1(i, \beta) = b_1^1$
- 14: $P_l^1(i, \beta) = b_2^1$
- 15: **end if**

is used to round down the value of $(\beta - w_i)$. The values of A^1 and B^1 denote the profit obtainable with and without new device in the list of operating ones, respectively. The case that gives better profit is then selected and the respective values are filled in the tables. The initial conditions of each tables are

$$\begin{aligned}
 P_e^1(0, \beta) &= -x, \beta = 0, \dots, R \\
 P_l^1(0, \beta) &= 0, \beta = 0, \dots, R \\
 P_e^1(i, 0) &= -x, i = 1, \dots, N \\
 P_l^1(i, 0) &= \sum_{j=1}^i \lambda \times l_{j0}^1, i = 1, \dots, N.
 \end{aligned}$$

After calculating all entries of the profit tables, a backtracking procedure, illustrated in Algorithm 8, is used to backtrack through the tables from the last cell to obtain the set of running devices with aid of the flag table F .

Let consider an example for three devices with the respective weight and operating probability: $(w_1, p_1) = (2, 0.8)$, $(w_2, p_2) = (5, 0.3)$, $(w_3, p_3) = (3.5, 0.9)$. With $P = 10$, $x = 5$, $\epsilon = 0.5$, the log-linear form of the operating probability of each device, i.e. $l_{i0} = -\log(1 - p_i)$ and $l_{i1} = -\log p_i, i = 1, \dots, 3$, are $(l_{10}, l_{11}) = (0.7, 0.1)$, $(l_{20}, l_{21}) = (0.15, 0.52)$, $(l_{30}, l_{31}) = (1, 0.05)$, respectively, and the boundary of total power R is equal to 6.

The entries of the profit tables are calculated by Algorithm 7 and filled in Table 4.1.

Algorithm 8 Backtracking algorithm in case of one-state devices.**Ensure:** State indicator vector $\mathbf{s} = \{s_1, \dots, s_N\}$

```

1:  $i = N$ 
2:  $\beta = R$ 
3: while  $i > 0$  and  $\beta > 0$  do
4:   if  $P_e^1(i, \beta) \neq P_e^1(i-1, \beta)$  or  $F(i, \beta) = 1$  then
5:      $s_i = 1$ 
6:      $\beta = \beta - \lceil w_i \rceil$ 
7:      $i = i - 1$ 
8:   else
9:      $s_i = 0$ 
10:     $i = i - 1$ 
11:   end if
12: end while

```

TABLE 4.1: Running example of DP in SmartSense for one-state devices

| $P_e^1(i, \beta)$ | | | | | | | |
|-------------------|------|------|------|------|------|------|------|
| (i, β) | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 0 | -5 | -5 | -5 | -5 | -5 | -5 | -5 |
| 1 | -5 | -5 | -3 | -3 | -3 | -3 | -3 |
| 2 | -5 | -5 | -3 | -3 | -3 | -3 | -3 |
| 3 | -5 | -5 | -3 | -3 | -1.5 | -1.5 | -0.5 |
| $P_l^1(i, \beta)$ | | | | | | | |
| (i, β) | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 7 | 7 | 1 | 1 | 1 | 1 | 1 |
| 2 | 8.5 | 8.5 | 2.5 | 2.5 | 2.5 | 2.5 | 2.5 |
| 3 | 18.5 | 18.5 | 12.5 | 12.5 | 9 | 9 | 3 |

The best profit is obtained at $(i, \beta) = (N, R)$. From this optimal point, the respective combination of running devices can be determined by backtracking through the table by applying Algorithm 8. As a result, $\mathbf{s} = (1, 0, 1)$, i.e. the first and third devices are operating while the second one is turned off. Apparently, this combination of devices gives the least absolute error on the power as well as the maximum operating probability.

Multi-state devices Each multi-state device is considered as a group of items, in which each state is an item with individual weight and profit. Similar to the case of one-state devices, two tables, P_e^M and P_l^M , are also constructed to contain the power and probability parameters of the profit. The algorithm to calculate the entries of the first table, $P_e^M(i, \beta) = \sum_{j=1}^i \sum_{m=1}^{m_i} (w_{jm} s_{jm} - x)$, and the second one, $P_l^M(i, \beta) = P \times \sum_{j=1}^i L^M(\mathbf{s}_j), i = 1, \dots, N, \beta = 1, \dots, R$, is detailed in Algorithm 9.

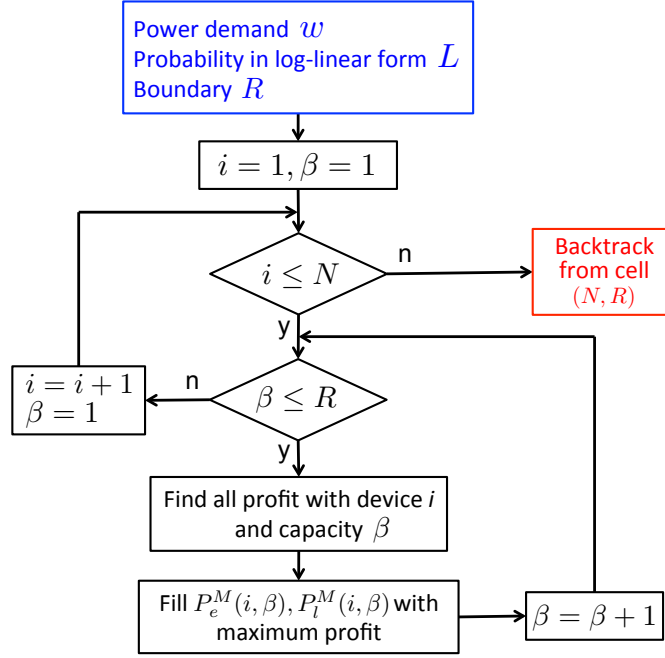


FIGURE 4.4: DP algorithm flowchart. When all entries are filled, the corresponding state vector is obtained by backtracking through the profit tables from the last cell (optimal value).

In other words, the value of each entry when considering a new device is determined by comparing the obtainable profit in two cases: the device is turned off, with profit A^M and the device is operating at a state satisfying the capacity β , with profit B^M . The initial conditions of the profit tables are

$$\begin{aligned}
 P_e^M(0, \beta) &= -x, \beta = 0, \dots, R \\
 P_l^M(0, \beta) &= 0, \beta = 0, \dots, R \\
 P_e^M(i, 0) &= -x, i = 1, \dots, N \\
 P_l^M(i, 0) &= \sum_{j=1}^i l_{j0}^M, i = 1, \dots, N.
 \end{aligned}$$

To get the state of each device, a backtracking procedure is also constructed as illustrated in Algorithm 10, in which an additional table *Sol* with the same size as the profit tables is used to mark the state of devices corresponding to each cell of profit tables.

Figure 4.4 illustrates the flowchart of the proposed DP algorithm in the general case with multi-state devices. Meanwhile, Table 4.2 shows the profit tables relating to the DP algorithm for three multi-state devices with the individual power demand $w_{11} = 2$, $(w_{21}, w_{22}) = (1, 4)$, $(w_{31}, w_{32}) = (3, 5)$ and the respective operating probability $p_{11} = 0.8$,

Algorithm 9 Entry calculation for profit tables in case of multi-state devices.

Ensure: $P_e^M(i, \beta), P_l^M(i, \beta)$

- 1: $Sol(i, \beta) = 0$
- 2: $a_1^M = P_e^M(i - 1, \beta)$
- 3: $a_2^M = P_l^M(i - 1, \beta) + \lambda \times l_{i0}^M$
- 4: $A^M = |a_1^M| + a_2^M$
- 5: **for** $j = 1, \dots, m_i$ **do**
- 6: **if** $w_{ij} \leq \beta$ **then**
- 7: $b_{j1}^M = P_e^M(i - 1, \lfloor \beta - w_{ij} \rfloor) + w_{ij}$
- 8: $b_{j2}^M = P_l^M(i - 1, \lfloor \beta - w_{ij} \rfloor) + \lambda \times l_{ij}^M$
- 9: $b_j^M = |b_{j1}^M| + b_{j2}^M$
- 10: **else**
- 11: $b_j^M = +\infty$
- 12: **end if**
- 13: **end for**
- 14: $k = \min_j \{b_j^M\}; B^M = b_k^M$
- 15: **if** $\min_j \{w_{ij}\} \leq \beta$ and $A^M \leq B^M$ **then**
- 16: $P_e^M(i, \beta) = a_1^M$
- 17: $P_l^M(i, \beta) = a_2^M$
- 18: **else**
- 19: $P_e^M(i, \beta) = b_{k1}^M$
- 20: $P_l^M(i, \beta) = b_{k2}^M$
- 21: $Sol(i, \beta) = k$
- 22: **end if**

Algorithm 10 Backtracking algorithm in case of multi-state devices.

Ensure: State indicator vector $\mathbf{s} = \{s_1, \dots, s_N\}$

- 1: $i = N$
- 2: $\beta = R$
- 3: **while** $i > 0$ and $\beta > 0$ **do**
- 4: **if** $Sol(i, \beta) > 0$ **then**
- 5: $k = Sol(i, \beta)$
- 6: $s_i = k$
- 7: $\beta = \beta - \lceil w_{ik} \rceil$
- 8: $i = i - 1$
- 9: **else**
- 10: $s_i = 0$ $i = i - 1$
- 11: **end if**
- 12: **end while**

TABLE 4.2: Running example of DP in SmartSense for multi-state devices

| $P_e^M(i, \beta)$ | | | | | | | |
|-------------------|----|------|------|-----|-----|-----|-----|
| (i, β) | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 0 | -6 | -6 | -6 | -6 | -6 | -6 | -6 |
| 1 | -6 | -6 | -4 | -4 | -4 | -4 | -4 |
| 2 | -6 | -5 | -5 | -3 | -3 | -3 | -3 |
| 3 | -6 | -5 | -5 | -3 | -3 | -3 | 0 |
| $P_l^M(i, \beta)$ | | | | | | | |
| (i, β) | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 7 | 7 | 1 | 1 | 1 | 1 | 1 |
| 2 | 17 | 8.5 | 8.5 | 2.5 | 2.5 | 2.5 | 2.5 |
| 3 | 24 | 15.5 | 15.5 | 9.5 | 9.5 | 9.5 | 4.7 |

$(p_{21}, p_{22}) = (0.7, 0.2)$, $(p_{31}, p_{32}) = (0.6, 0.2)$. The aggregate power consumption x in this example is equal to 6 with standard deviation $\epsilon = 0$ and the respective total power boundary $R = \lceil x + \epsilon \rceil = 6$. The regularization parameter P is set to be of 10. Applying the log-linear form to the operating probability, i.e. $l_{ij} = -\log p_{ij}$, $i = 1, \dots, 3$, $j = 0, \dots, m_i$, the corresponding results are $(l_{10}, l_{11}) = (0.7, 0.1)$, $(l_{20}, l_{21}, l_{22}) = (1, 0.15, 0.7)$, $(l_{30}, l_{31}, l_{32}) = (0.7, 0.22, 0.7)$. Each entry of these tables is calculated by Algorithm 9 and the set of operating state of each device is backtracked through the tables from the optimal point, i.e. $(i, \beta) = (N, R)$, by applying Algorithm 10 is then $\mathbf{s} = (1, 1, 1)$. This result means that all three devices are operating at the first power state. This determination also satisfies the requirement on the least absolute error on the power consumption and the maximum coincidence operating probability. Presenting vector \mathbf{s} as in Eq. 4.16, we have $\mathbf{s} = \{1, 1, 0, 1, 0\}$.

4.3 Approach 2: Edge Detector Algorithms

The edge detector was firstly proposed in [1] to detect the step-changes on the power signal. Both ED and DTW algorithms at first try to detect the rising edge and falling edge of a device activation and pair them together. Denote $\Delta x(t) = x(t) - x(t-1)$, a rising edge is detected at time t if $\Delta x(t) \geq \gamma$, where γ is empirically chosen so that it is small enough not to miss the edges, but large enough to ignore the variations in a stable period. In contrast, a falling edge is detected at time $(t-1)$ if $\Delta x(t) \leq -\gamma$. A rising edge and a falling edge are paired together if the difference between their height is lower than

a threshold α , i.e.,

$$|\Delta x(t_s) - \Delta x(t_e)| \leq \alpha. \quad (4.21)$$

In the case of multi-state devices, it appears more than one consecutive rising edges or falling edges in an activation. These edges will be grouped together if

$$\left| \sum_{i=1}^{n_s} \Delta x(t_s^i) - \sum_{i=1}^{n_e} \Delta x(t_e^i) \right| \leq \alpha, \quad (4.22)$$

where n_s , n_e are the number of consecutive rising edges and falling edges, respectively.

In the ED algorithm, a feature is composed of the first rising edge and the last falling edge, while all active power values between them are used in DTW feature. To improve the performance of the edge detector, a median filter can also be applied to remove the peaks [44, 45].

4.3.1 Edge Detection (ED)

In training period, all possible patterns of each device are extracted and saved in the library **EDlib**, in which pattern j of device i is denoted as

$$f_{ij}^{ed} = \{\Delta x_{ij}^R, \Delta x_{ij}^F\}, \quad (4.23)$$

with notation R for rising edge and F for falling edge. If the edge detector detects an event switched on at time t_s and switched off at time t_e , the corresponding feature is

$$f^{ed} = \{\Delta x(t_s^1), \Delta x(t_e^{n_e})\}. \quad (4.24)$$

The distance between two patterns f_{ij}^{ed} and f^{ed} can be calculated as

$$\begin{aligned} \delta_{ij}^{ed} &= d(f^{ed}, f_{ij}^{ed}) \\ &= |\Delta x(t_s^1) - \Delta x_{ij}^R| + |\Delta x(t_e^{n_e}) - \Delta x_{ij}^F|. \end{aligned} \quad (4.25)$$

In the ED algorithm, the detected event will be matched with the device having a pattern with the minimum distance. Nevertheless, in homes and buildings, there are several devices with the same edge height, which decreases the load disaggregation performance. To overcome this challenge, in the SmartSense system, the probability that each device

$i, i = 1, \dots, N$, is turned on and off at the same time as the detected event, denoted as $p_i^R(t_s^1)$ and $p_i^F(t_e^{n_e})$, respectively, is estimated as in Section 4.1. This means, $p_i^R(t_s^1) = (1 - npv)$ if a rising edge of device i is detected at time t_s^1 and $p_i^R(t_s^1) = (1 - npv)$ on the contrary. Similarly, $p_i^F(t_e^{n_e}) = pr$ if device i has a falling edge at time $t_e^{n_e}$ and $p_i^F(t_e^{n_e}) = (1 - npv)$ on the contrary. In the case of unsupervised devices, $p_i^R(t_s^1) = p_i^F(t_e^{n_e}) = 0.5$. The coincidence probability that both edges appear is then

$$Pr_i^{ed} = p_i^R(t_s^1) \times p_i^F(t_e^{n_e}). \quad (4.26)$$

To reduce the computational complexity, the multiplication in Eq. (4.26) can be transformed to an addition by applying the log-linear form, which gives

$$\begin{aligned} \phi_i^{ed} &= -\log Pr_i^{ed} \\ &= -\log p_i^R(t_s^1) - \log p_i^F(t_e^{n_e}). \end{aligned} \quad (4.27)$$

The identified device does not only have a pattern with the minimum distance δ_{ij}^{ed} , but also has the maximum probability Pr_i^{ed} , i.e. the minimum log-linear form ϕ_i^{ed} . In other words, the ED algorithm will try to find the pattern giving the minimum modified distance, calculated as

$$d_{ij}^{ed} = \delta_{ij}^{ed} + \lambda_1 \times \phi_i^{ed}, \quad (4.28)$$

where λ_1 is a regularization parameter. The load identification procedure for each detected pattern is presented in Algorithm 11, where Φ^{ed} contains the probability in log-linear form of all devices and m_i denotes the number of patterns of device i in **EDlib**. The procedure is illustrated by a flowchart in Figure 4.5.

4.3.2 Dynamic Time Warping (DTW)

Different from the ED algorithm, in DTW, all active power values between the first rising edge and the last falling edge are extracted to comprise a pattern, i.e. $f^{dtw} = \{f^{dtw}(k) | k =$

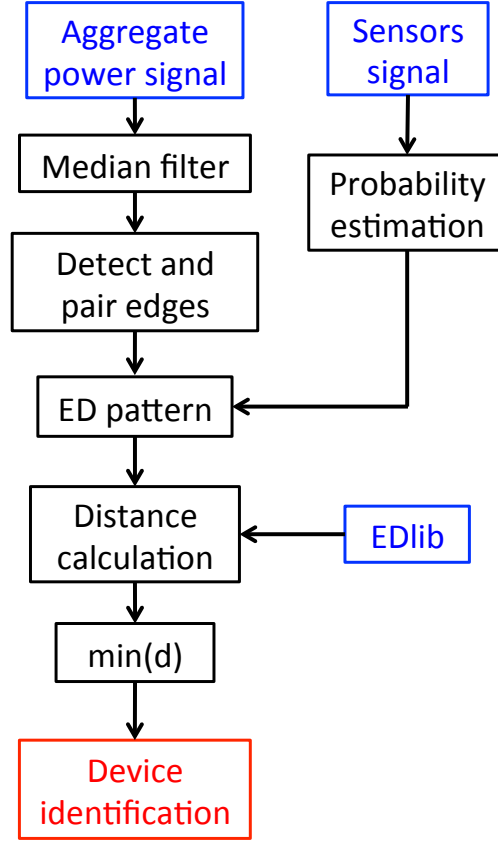


FIGURE 4.5: Load disaggregation based on the ED algorithm.

$1, \dots, t_e^{n_e} - t_s^1 + 1\}$, which

$$f^{dtw}(k) = \begin{cases} h_s^i, & z_s^i \leq k < z_s^{i+1}, 1 \leq i \leq n_s - 1 \\ h_s^{n_s} + \frac{h_e^1 - h_s^{n_s}}{z_e^1 - z_s^{n_s}} (k - z_s^{n_s}), & z_s^{n_s} \leq k \leq z_e^1 \\ h_s^{i+1}, & z_e^i < k \leq z_e^{i+1}, 1 < i \leq n_e - 1 \end{cases} \quad (4.29)$$

where $h_s^i = \sum_{j=1}^i \Delta x(t_s^j)$, $h_e^i = \left| \sum_{j=i}^{n_e} \Delta x(t_e^j) \right|$, $z_s^i = t_s^i - t_s^1 + 1$, and $z_e^i = t_e^i - t_s^1 + 1$. Figure 4.6 illustrates a DTW pattern with two rising edges and two falling edges. All possible patterns of each device are extracted and saved in library **DTWlib** from the training period. Because the length of patterns is different, the accumulated distance between the detected pattern f^{dtw} and a pattern j of device i in the library is calculated by Algorithm 12. Concretely,

$$\delta_{ij}^{dtw} = \text{AccDistance}(f^{dtw}, f_{ij}^{dtw}). \quad (4.30)$$

Algorithm 11 Matching detected event in ED.

```

1: function EDIDEN( $f^{ed}, \Phi^{ed}, \mathbf{EDlib}$ )
2:    $MIN = +\infty$ 
3:    $k = 0$ 
4:   for  $i = 1, \dots, N$  do
5:      $\phi_i^{ed} = \Phi^{ed}(i)$ 
6:     for  $j = 1, \dots, m_i$  do
7:        $f_{ij}^{ed} = \mathbf{EDlib}\{i, j\}$ 
8:        $\delta_{ij}^{ed} = |\Delta x(t_s^1) - \Delta x_{ij}^R| + |\Delta x(t_e^e) - \Delta x_{ij}^F|$ 
9:        $d_{ij}^{ed} = \delta_{ij}^{ed} + \lambda_1 \times \phi_i^{ed}$ 
10:      if  $d_{ij}^{ed} \leq MIN$  then
11:         $MIN = d_{ij}^{ed}$ 
12:         $k = i$ 
13:      end if
14:    end for
15:  end for
16:   $\text{output} = k$ 
17: end function

```

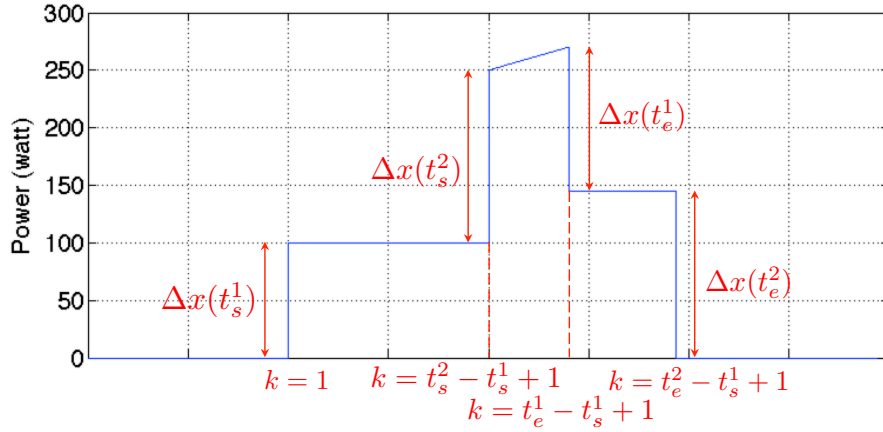


FIGURE 4.6: DTW pattern.

In [2], the detected pattern will be matched to the device having a pattern with minimum accumulated distance calculated by Eq. (4.30). However, in SmartSense, in order to improve the performance, the operating probability of each device during the same period as the detected event will also be used in log-linear form, and is defined as

$$Pr_i^{dtw} = \prod_{t=t_s^1}^{t_e^{ne}} p_i(t) \quad (4.31)$$

$$\begin{aligned} \phi_i^{dtw} &= -\log Pr_i^{dtw} \\ &= -\sum_{t=t_s^1}^{t_e^{ne}} \log p_i(t), \end{aligned} \quad (4.32)$$

where $p_i(t)$ is the on-state probability of device i at time instant t , estimated as in

Algorithm 12 Accumulated distance between two different length vectors [2].

```

1: function ACCDISTANCE( $f1, f2$ )
2:    $l1 = \text{length}(f1)$ 
3:    $l2 = \text{length}(f2)$ 
4:    $\delta(0, 0) = 0; \delta(i, 0) = \delta(0, i) = +\infty$ 
5:   for  $i = 1, \dots, l1$  do
6:     for  $j = 1, \dots, l2$  do
7:        $d(i, j) = |f1(i) - f2(j)|$ 
8:        $\delta(i, j) = d(i, j) + \min \{ \delta(i-1, j), \delta(i-1, j-1), \delta(i, j-1) \}$ 
9:     end for
10:  end for
11:   $\text{output} = \delta(l1, l2)$ 
12: end function

```

Section 4.1, i.e. $p_i(t) = pr$ if device i is detected as on and $p_i(t) = (1 - npv)$ if detected as off. In the case of unsupervised devices, $p_i(t) = 0.5$. The load identification, as presented in Algorithm 13, is not only based on the accumulated distance as in Eq. (4.30), but also considers the operating probability of each device in Eq. (4.32). The probability in log-linear form of all devices is contained in vector Φ^{dtw} . A modified distance then combines these two parameters with a regularization parameter λ_2 , which gives

$$d_{ij}^{dtw} = \delta_{ij}^{dtw} + \lambda_s \times \phi_i^{dtw}. \quad (4.33)$$

The pattern giving the minimum distance will be selected to identify the corresponding device. Because the length of patterns strongly affects the accumulated distance, the DTW algorithm is only suitable for devices with fixed on-duration, such as fridge, washing machine, dish washer. Other devices controlled by the human intervention, e.g. lamp, computer, etc., cannot be exactly identified. Therefore, we propose to combine DTW with ED, in which **DTWlib** only contains the patterns of fixed on-duration devices and **EDlib** saves the edge patterns of the others. The flowchart of the proposed DTW algorithm is shown in Figure 4.7. A threshold β is used to reject a DTW pattern if it does not match with any one in **DTWlib**. The detected pattern will be then extracted based only on the first rising edge and the last falling edge, by applying the ED procedure.

Algorithm 13 Matching detected event in DTW.

```

1: function DTWIDEN( $f^{dtw}, \Phi^{dtw}, \mathbf{DTWlib}$ )
2:    $MIN = +\infty$ 
3:    $k = 0$ 
4:   for  $i = 1, \dots, N$  do
5:      $\phi_i^{dtw} = \Phi^{dtw}(i)$ 
6:     for  $j = 1, \dots, m_i$  do
7:        $f_{ij}^{dtw} = \mathbf{DTWlib}\{i, j\}$ 
8:        $\delta_{ij}^{dtw} = \text{AccDistance}(f^{dtw}, f_{ij}^{dtw})$ 
9:        $d_{ij}^{dtw} = \delta_{ij}^{dtw} + \lambda_2 \times \phi_i^{dtw}$ 
10:      if  $d_{ij}^{dtw} \leq \beta$  and  $d_{ij}^{dtw} \leq MIN$  then
11:         $MIN = d_{ij}^{dtw}$ 
12:         $k = i$ 
13:      end if
14:    end for
15:  end for
16:  output =  $k$ 
17: end function

```

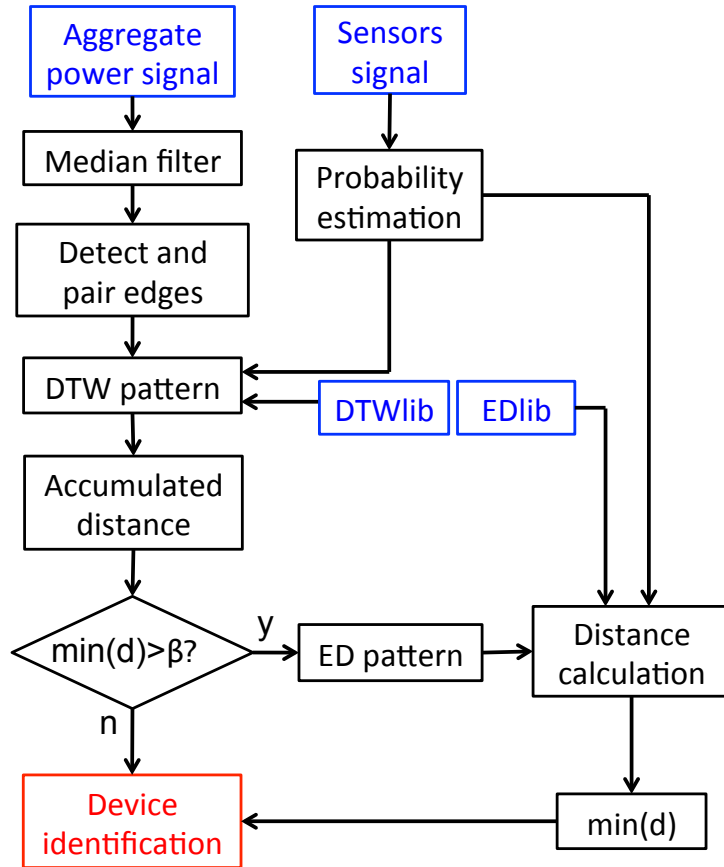


FIGURE 4.7: Combination of DTW and ED algorithms. A threshold β is used to reject a pattern if it does not match with any one in **DTWlib**

Chapter 5

Experimental Results on SmartSense

5.1 Dataset and Metrics

In the context of this thesis, the sensor signal processing to extract the state of devices is not studied nor implemented. We just focus on how to transform this information to probability feature to help the existing NILM algorithms to improve their performance. Therefore, with each dataset applied to the simulation in Matlab, we assume a WSN monitoring some specific devices. The operation of these devices can be detected with a certain accuracy. To evaluate the WSN performance, besides the precision, recall, F-measure introduced in Chapter 3 and negative predictive value as in Section 4.1, another metric called *true negative rate* (tnr) is also used and calculated as follows:

$$tnr = \frac{TN}{TN + FP}. \quad (5.1)$$

Denote r as the ratio between the total events and total non-events of the data, the true negative rate and negative predictive value are calculated as functions of precision and recall, such as:

$$tnr = 1 - \frac{rc \times r \times (1 - pr)}{pr} \quad (5.2)$$

$$npv = 1 - \frac{pr \times r \times (1 - rc)}{pr \times (1 + r) - rc \times r}. \quad (5.3)$$

TABLE 5.1: List of devices in REDD 1 and their characteristics.

| Label | Device | Number of uses | Using rate (%) | Power (watts) |
|-------|-----------------|----------------|----------------|--------------------|
| 1 | Oven | 153 | 0.24 | {4140, 4075, 3377} |
| 2 | Fridge | 1695 | 24.94 | {193, 423} |
| 3 | Dish washer | 383 | 3.80 | {214, 1107} |
| 4 | Lighting | 67 | 48.99 | {169, 275, 345} |
| 5 | Microwave | 522 | 1.94 | {1531, 1571, 1396} |
| 6 | Outlet bathroom | 226 | 0.39 | 1595 |
| 7 | Outlet | 189 | 0.38 | 1056 |
| 8 | Outlet | 90 | 0.2 | 1521 |
| 9 | Washer dryer | 234 | 0.76 | {5097, 5250, 5008} |

Knowing the characteristics of the sensor network, the data of individual circuits can be modified to apply in SmartSense. Whereby, $(1 - rc)$ of events are randomly selected and changed to non-events, and in contrast, $(1 - tnr)$ of non-events are randomly selected and changed to events. At these instants, the corresponding devices are assumed to be incorrectly detected. As a consequence, the operating probability of each device will be determined from the modified data based on the precision and negative predictive value as presented in Section 4.1.

In the simulation, the proposed algorithms are applied to four datasets: UK-DALE House 5 (UK-DALE 5) [55], REDD House 1 (REDD 1) and House 2 (REDD 2) [4], and our Athemium dataset retrieved from the coffee room as introduced in Chapter 3. The devices as well as their characteristics in each dataset are detailed in Tables 5.1, 5.2, 5.3 and 5.4, respectively. Both UK-DALE and REDD public datasets are retrieved by measuring the power consumption at the main power line and of several typical devices in home sampled every 3-4 seconds. Although there are over 20 channels in each dataset, we use only a part of them because the others have no activation and the variable loads are ignored. In addition, the data from two main phases in REDD dataset are also combined to have a unique main power signal. The data in each dataset are divided into two parts: the first one (30%) to train the power characteristics of each device and the second one (70%) to test the algorithms. During the training period, the *ghost power*, coming from the unsupervised devices, is assumed to be consumed by a virtual device.

The simulation with the training data also allows to empirically choose the value of the regularization parameters, as shown in Table 5.5. The training dataset is divided into two

TABLE 5.2: List of devices in REDD 2 and their characteristics.

| Label | Device | Number of uses | Using rate (%) | Power (watts) |
|-------|-------------|----------------|----------------|---------------|
| 1 | Outlet | 180 | 0.25 | 779 |
| 2 | Lighting | 275 | 12.67 | {153, 71} |
| 3 | Stove | 63 | 0.22 | 406 |
| 4 | Microwave | 120 | 0.69 | {1887, 1778} |
| 5 | Outlet | 126 | 0.86 | 1058 |
| 6 | Fridge | 2715 | 44.40 | {163, 426} |
| 7 | Dish washer | 207 | 1.19 | {244, 1214} |

TABLE 5.3: List of devices in UK-DALE 5 and their characteristics.

| Label | Device | Number of uses | Using rate (%) | Power (watts) |
|-------|--------------|----------------|----------------|---------------------|
| 1 | Toaster | 162 | 0.06 | 1088 |
| 2 | Kettle | 1013 | 0.37 | {2826, 2960} |
| 3 | Fridge | 8312 | 36.48 | 109.5 |
| 4 | Oven | 4268 | 22.56 | {110.6, 2119, 2933} |
| 5 | Electric hob | 2260 | 0.52 | {961, 1878, 1611} |
| 6 | Dish washer | 365 | 2.38 | {1660, 96.3, 57.4} |
| 7 | Microwave | 186 | 68.40 | {1496, 1258, 49.8} |
| 8 | Washer dryer | 5942 | 2.89 | {61.7, 1492, 290.6} |

TABLE 5.4: List of devices in Athemium dataset and their characteristics.

| Label | Device | Number of uses | Using rate (%) | Power (watts) |
|-------|----------------|----------------|----------------|-----------------|
| 1 | Fridge | 3040 | 17.79 | 75 |
| 2 | Coffee machine | 1345 | 1.57 | {823, 798, 234} |
| 3 | Teapot | 640 | 0.94 | {1635, 1692} |
| 4 | Microwave | 250 | 0.24 | {1290, 1349} |
| 5 | Television | 70 | 35.76 | {203, 29} |
| 6 | Monitor | 110 | 27.35 | 73 |

parts: the first part (80%) is applied to learn the power characteristics of each device and the second part, also called developing set, is used to train the regularization parameters.

TABLE 5.5: Regularization parameters.

| | α | β | γ | δ | λ | ϵ | L |
|-----|----------|---------|----------|----------|-----------|------------|-----|
| CPH | | | | | 60 | 20 | 128 |
| DP | | | | 10 | 60 | 20 | |
| DTW | | 500 | 60 | | 2 | | |
| ED | 30 | | 60 | | 10 | | |

TABLE 5.6: Performance of the algorithms without monitored device (%).

| | REDD 1 | | | REDD 2 | | |
|-----|--------|--------|--------|--------|--------|--------|
| | pr_0 | rc_0 | Fm_0 | pr_0 | rc_0 | Fm_0 |
| CPH | 61.09 | 65.12 | 63.04 | 66.64 | 74.21 | 70.22 |
| DP | 61.34 | 65.11 | 63.17 | 67.18 | 73.49 | 70.19 |
| DTW | 83.53 | 76.48 | 79.85 | 83.77 | 76.50 | 79.97 |
| ED | 80.95 | 70.58 | 75.41 | 83.71 | 68.32 | 75.24 |

| | UK-DALE 5 | | | Athemium | | |
|-----|-----------|--------|--------|----------|--------|--------|
| | pr_0 | rc_0 | Fm_0 | pr_0 | rc_0 | Fm_0 |
| CPH | 66.88 | 59.19 | 62.80 | 65.51 | 71.25 | 68.26 |
| DP | 68.67 | 59.48 | 63.74 | 65.92 | 71.10 | 68.41 |

5.2 Results and Evaluation of Probability Information in SmartSense

5.2.1 Performance without probability information

To evaluate the effect of the probability information on the performance improvement of NILM algorithms, we consider the gain obtained by comparing the overall performance of each algorithm without monitored device. The performance of the algorithms without probability information are provided in Table 5.6. In an original NILM system, both DTW and ED algorithms show a better performance than the CPH and DP ones. Not only that, in [2], DTW is proved to outperform other state-of-the-art approaches when applied to REDD dataset.

Additionally, the performance per device is also considered and compared with the traditional NILM system given in Tables 5.7, 5.8, 5.9 and 5.10. Apparently, the performance of the DTW algorithm in terms of F-measure related to the devices with fixed on-duration is improved compared with the ED one. For example in REDD 1, the fridge (device 2) is

TABLE 5.7: Performance per device in REDD 1 without monitored device (%).

| Device | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| CPH | pr_0 | 26.49 | 62.13 | 23.83 | 77.02 | 26.01 | 19.16 | 13.14 | 24.46 | 80.25 |
| | rc_0 | 49.71 | 57.78 | 69.28 | 71.86 | 16.07 | 20.52 | 21.57 | 18.14 | 70.67 |
| | Fm_0 | 34.57 | 59.88 | 35.48 | 74.35 | 19.86 | 19.82 | 16.33 | 20.83 | 75.16 |
| DP | pr_0 | 48.27 | 58.62 | 24.09 | 73.49 | 25.87 | 15.84 | 18.22 | 13.82 | 90.31 |
| | rc_0 | 35.61 | 34.80 | 70.23 | 71.85 | 40.51 | 24.19 | 23.17 | 35.71 | 71.17 |
| | Fm_0 | 40.99 | 43.67 | 35.87 | 72.66 | 31.58 | 19.14 | 20.40 | 19.92 | 79.61 |
| ED | pr_0 | 42.88 | 89.70 | 79.47 | 34.85 | 28.85 | 40.52 | 49.27 | 19.16 | 90.75 |
| | rc_0 | 59.66 | 78.32 | 51.48 | 25.05 | 58.01 | 61.90 | 43.21 | 17.45 | 54.18 |
| | Fm_0 | 49.89 | 83.62 | 62.48 | 29.15 | 38.53 | 48.98 | 46.04 | 18.26 | 67.85 |
| DTW | pr_0 | 47.54 | 85.39 | 78.85 | 68.81 | 34.17 | 39.74 | 49.27 | 21.46 | 99.36 |
| | rc_0 | 74.95 | 83.49 | 56.19 | 25.05 | 68.01 | 61.90 | 43.21 | 20.45 | 67.61 |
| | Fm_0 | 58.18 | 84.43 | 65.62 | 36.73 | 45.48 | 48.40 | 46.04 | 20.94 | 80.47 |

TABLE 5.8: Performance per device in REDD 2 without monitored devices (%).

| Device | | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|--------|--------|-------|-------|-------|-------|-------|-------|-------|
| CPH | pr_0 | 19.72 | 26.07 | 26.17 | 90.98 | 29.05 | 82.40 | 25.64 |
| | rc_0 | 22.63 | 37.09 | 50.87 | 16.13 | 25.62 | 90.46 | 51.05 |
| | Fm_0 | 21.08 | 30.62 | 34.56 | 27.40 | 27.23 | 86.24 | 34.14 |
| DP | pr_0 | 17.86 | 26.94 | 23.93 | 90.52 | 24.86 | 89.06 | 24.43 |
| | rc_0 | 18.37 | 25.06 | 57.10 | 11.46 | 23.08 | 80.64 | 48.07 |
| | Fm_0 | 18.11 | 25.97 | 33.73 | 20.34 | 23.94 | 84.64 | 32.40 |
| ED | pr_0 | 80.65 | 93.81 | 21.55 | 98.65 | 60.09 | 67.60 | 21.08 |
| | rc_0 | 19.65 | 40.04 | 95.13 | 17.62 | 75.12 | 83.96 | 45.66 |
| | Fm_0 | 31.60 | 56.12 | 35.15 | 29.90 | 68.67 | 71.54 | 28.83 |
| DTW | pr_0 | 80.65 | 94.17 | 21.55 | 52.80 | 63.09 | 80.57 | 27.82 |
| | rc_0 | 19.65 | 39.01 | 93.13 | 39.04 | 85.12 | 84.11 | 55.15 |
| | Fm_0 | 31.60 | 55.17 | 35.01 | 44.89 | 72.47 | 82.30 | 36.98 |

TABLE 5.9: Performance per device in UK-DALE 5 without monitored device (%).

| Device | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| CPH | pr_0 | 11.76 | 32.82 | 73.26 | 26.39 | 14.67 | 6.10 | 86.61 | 75.25 |
| | rc_0 | 51.02 | 38.87 | 27.04 | 21.04 | 44.28 | 54.05 | 65.34 | 62.82 |
| | Fm_0 | 19.11 | 35.59 | 39.50 | 23.41 | 22.04 | 10.97 | 74.49 | 68.48 |
| DP | pr_0 | 6.23 | 39.96 | 82.27 | 24.17 | 14.91 | 3.85 | 83.76 | 88.50 |
| | rc_0 | 23.62 | 53.23 | 82.42 | 7.30 | 52.20 | 50.40 | 53.90 | 65.02 |
| | Fm_0 | 9.86 | 45.65 | 82.34 | 11.22 | 23.20 | 7.16 | 65.59 | 74.97 |

TABLE 5.10: Performance per device in Athemium without monitored device (%).

| Device | | 1 | 2 | 3 | 4 | 5 | 6 |
|--------|--------|-------|-------|-------|-------|-------|-------|
| CPH | pr_0 | 48.30 | 79.78 | 96.95 | 54.51 | 83.84 | 72.32 |
| | rc_0 | 94.09 | 99.05 | 44.92 | 76.22 | 97.24 | 52.47 |
| | Fm_0 | 63.83 | 88.38 | 61.39 | 63.56 | 90.04 | 60.82 |
| DP | pr_0 | 47.75 | 89.34 | 99.44 | 47.02 | 82.29 | 85.40 |
| | rc_0 | 97.11 | 98.98 | 62.22 | 83.99 | 97.26 | 57.78 |
| | Fm_0 | 64.02 | 93.91 | 76.55 | 60.29 | 89.15 | 68.93 |

detected with a precision of 89.7% and recall of 78.32% by the ED algorithm. Although the DTW method can only identify it with a lower precision (85.39%), the significant improvement of the recall (from 78.32% to 83.49%) allows to increase the F-measure from 83.62% to 84.43%. Meanwhile, the performance related to the washer dryer (9) is improved from 90.75% of precision and 54.18% of recall by ED algorithm to 99.36% and 67.61%, respectively, by the DTW algorithm. Similarly, the detection of the fridge (6) in REDD 2 increases from the precision of 67.6% and recall of 83.96% up to 80.57% and 84.11%, respectively. In Knapsack approach, because of directly using the power consumption as feature, the performance of the corresponding algorithms is strongly affected by the power demand of each device. The devices with confusion on the power consumption with the others, especially one-state devices, are detected with low accuracy. For example, the device connected to the outlet 8 (1521W) in REDD 1 has a very closed power demand to the microwave (1531W) and the outlet bathroom (1595W). This is the reason why the algorithms detect its operation with a precision and recall lower than 35%. Similarly, in REDD 2, the power consumption of the dish washer (1214W) is incorrectly disaggregated to the combination of outlet 1 (779W) and stove (406W). Obviously, if one of these devices is monitored, not only this device but also other confused devices will be more correctly identified and the overall performance can be improved. In the next section, we will analyze the elements affecting the performance of the proposed algorithms in the two proposed approaches: knapsack and edge detector.

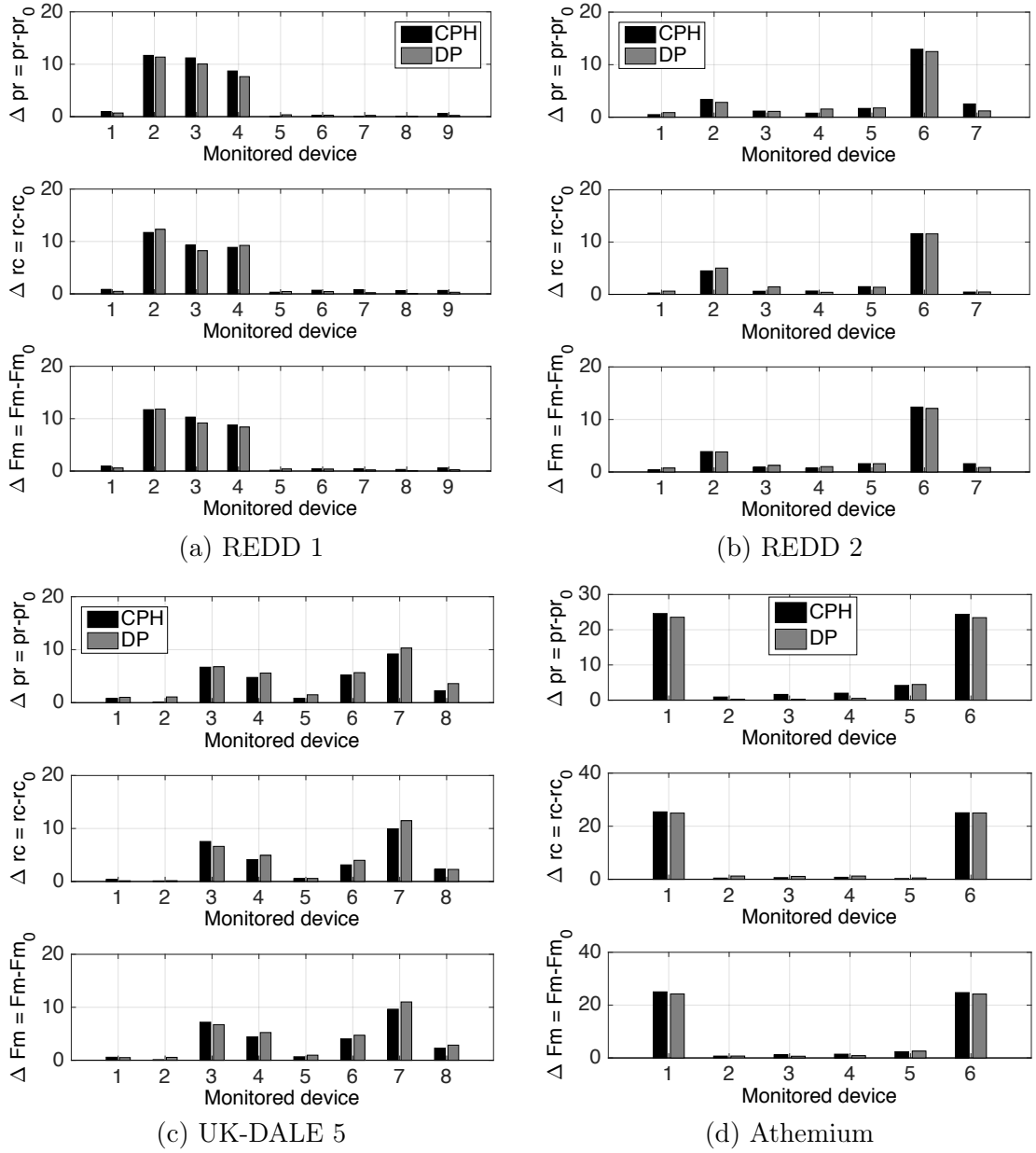


FIGURE 5.1: Effect of the type of monitored devices on the performance improvement of the CPH and DP algorithms. The values of pr_0 , rc_0 , Fm_0 correspond to the case without any monitored device, given in Table 5.6.

5.2.2 Impact of type and number of monitored devices in the Knapsack approach

In SmartSense, only several specific devices are selected to be monitored, but a relevant question is: Which type of devices can help to improve the performance of the algorithms? In the first experiment, sequentially each device is selected to monitor, the main characteristics of the devices giving the best performance are then analyzed to generalize

the criteria to select the monitored devices. The precision and recall of the sensors detection are assumed to be equal to 0.9. Figure 5.1 shows the performance of the CPH and DP algorithms with different monitored devices in four dataset, in which the labels are respectively noted in Tables 5.1, 5.2, 5.3 and 5.4. Because both CPH and DP are searching for the optimal solution, they give similar results. Table 5.11 lists two devices with the best performance in each dataset. In REDD 1, as shown in Figure 5.1(a), the best performance corresponds to the fridge with the gain of both precision and recall around 12%. From Table 5.1, it can be seen that the fridge has a high using rate (24.94%) and its power demand is confused with the dish washer (193W *vs.* 214W). Two other devices also giving a high gain are the dish washer and the lighting system. The lighting system has the largest using rate (48.99%). Although unusually operating, monitoring the dish washer also gives a high performance gain because it consumes a power similar to the fridge. Meanwhile, other devices show a very small gain ($<2\%$) if monitored. In Figure 5.1(b), except for the fridge with the performance improvement over 12% for all evaluation metrics, deploying the sensors to monitor other devices in REDD 2 can only obtain a gain lower than 5%. This result can be explained from Table 5.2 that besides the most used device (44.40%), the fridge consumes a power very close to the lighting system (163W *vs.* 153W) and stove (426W *vs.* 406W).

Therefore, we can come to a conclusion that the most effective devices to be monitored with CPH and DP algorithms have two main characteristics that are: high using rate and confusion on the power demand with other devices. This conclusion is also consolidated by the results from the UK-DALE 5 and Athemium dataset. As shown in Figure 5.1(c), the most effective devices in UK-DALE 5 are the microwave and fridge, which both have high using ratio (68.4% and 36.48% respectively) and their power demand is also confused with others. Concretely, the microwave consumes a power close to the washer dryer (1496W and 1492W) and dish washer (49.8W and 57.4W), while the power demand of the fridge is near to the dish washer (109.5W and 96.3W). Meanwhile, in Figure 5.1(d), the most effective devices in Athemium dataset are fridge and monitor, which are frequently used (17.79% and 27.35% respectively) and their power consumption is also similar (75W and 73W).

Besides the type, the number of monitored devices is also a parameter strongly affecting on the performance improvement of the algorithms. Apparently, monitoring more devices allows to obtain a better performance. However, the deployment cost also increases along

TABLE 5.11: Most effective devices with the CPH and DP algorithms.

| | CPH | DP |
|-----------|---------------------------|---------------------------|
| REDD 1 | 2-fridge 3-dish washer | 2-fridge 3-dish washer |
| REDD 2 | 6-fridge 2-lighting | 6-fridge 2-lighting |
| UK-DALE 5 | 7-microwave 3-fridge | 7-microwave 3-fridge |
| Athemium | 1-fridge 6-monitor | 1-fridge 6-monitor |

with the size of the monitoring sensors network and the non-intrusive requirement is violated. Therefore, it is necessary to make a trade-off between the size of the WSN and the desired performance when selecting the monitored devices. Figure 5.2 shows the performance improvement when increasing the number of monitored devices. In each case, the devices are selected based on their performance in Figure 5.1. Obviously, there are some devices showing less effectiveness when monitored and adding them to the monitoring list cannot help to significantly improve the performance. Thus, these devices can be ignored when deploying the sensors network. Concretely, in REDD 1, three devices giving the best performance are fridge, dish washer and lighting system. Thus, adding these devices to the monitoring list allows to significantly improve the performance of the algorithms. This phenomenon can be illustrated by the slope from one to three devices in Figure 5.2(a). The corresponding precision gain increases from 12% to 30% and the recall from 11% to 19% in this interval. However, other devices are less effective, which results in a slow increase of the performance gain with more than three monitored devices (from 30% to 37% with precision and 19% to 23% with recall when increasing the number of devices from three to nine). Similarly, in REDD 2, as shown in Figure 5.2(b), the performance is improved very slowly. After monitoring the first device (fridge) and obtaining the precision gain of 13% and recall gain of 12%, we can only improve the recall around 3% (from 12% to 15%) with other six devices added to the monitoring list. Therefore, the F-measure is still insignificant (11% improved) although the precision gain can be improved from 13% up to 31%. Meanwhile, in UK-DALE 5, there are two devices giving a remarkable increase of the performance when monitored including the microwave and fridge, as shown in Figure 5.1(c). This is the reason why the performance is significantly improved with the number of monitored devices from one to two, and

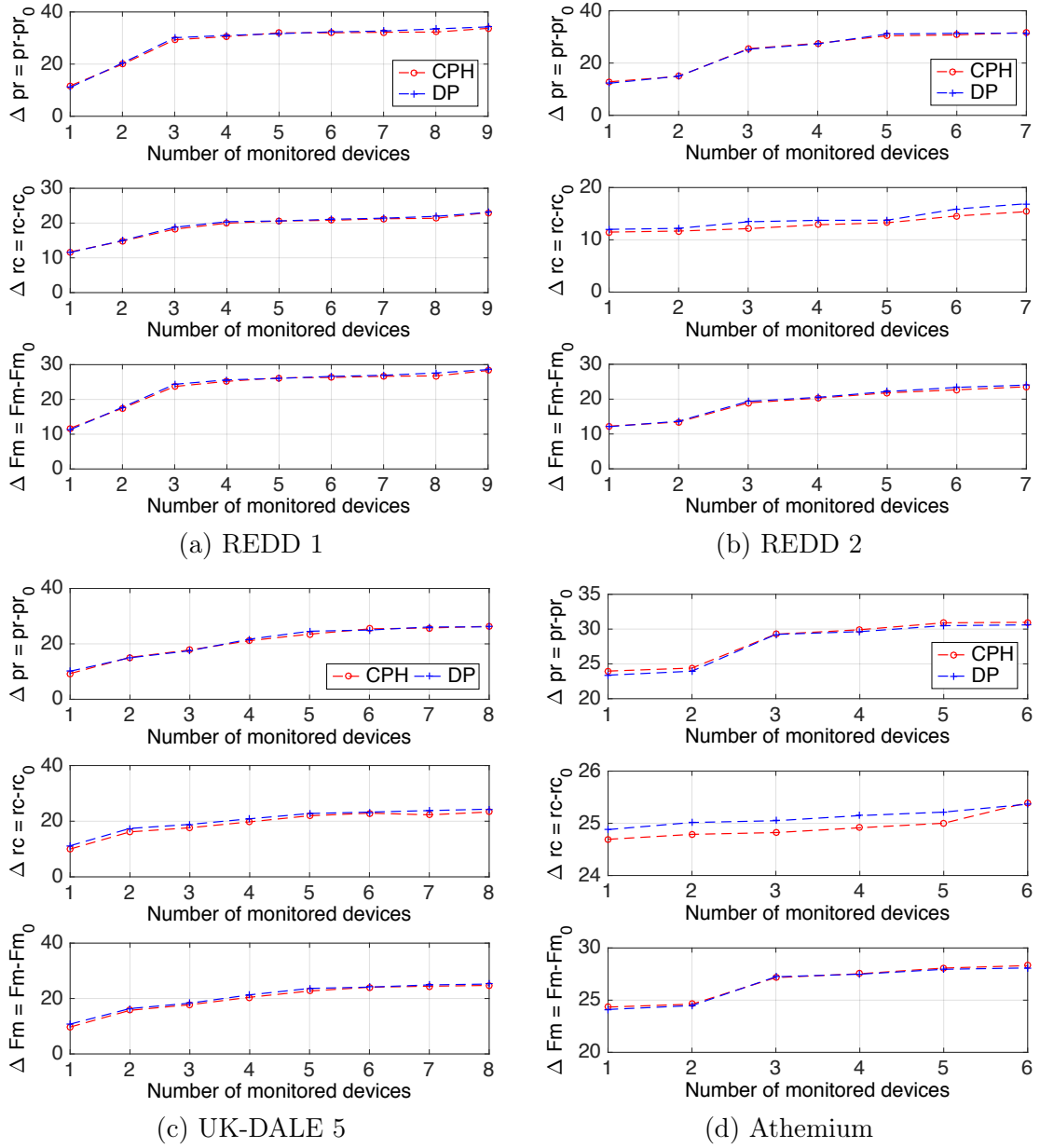
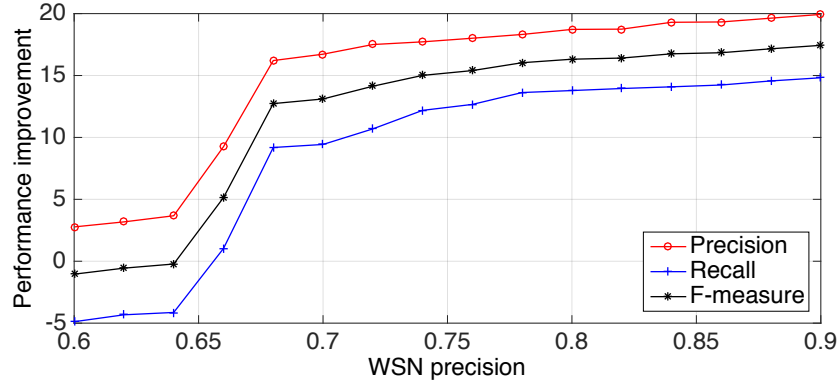
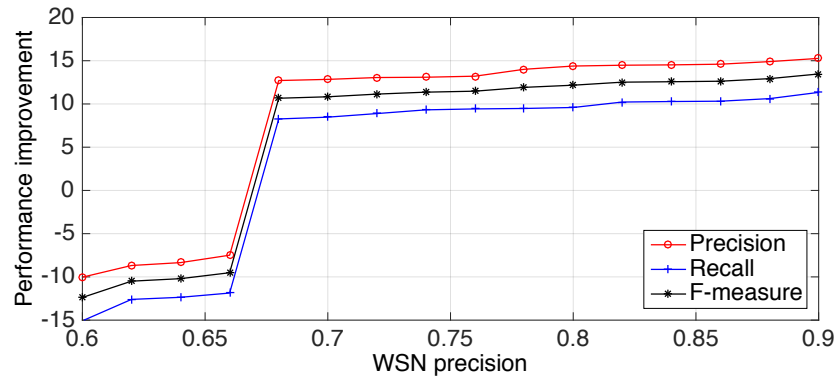


FIGURE 5.2: Effect of the number of monitored devices on the performance improvement of the CPH and DP algorithms. The values of pr_0 , rc_0 , Fm_0 correspond to the case without any monitored device, given in Table 5.6.

slowly increased with more other devices. In contrast, in Athemium dataset, because the fridge and the monitor have only one power state and their power demand are confused together, the performance can be significantly improved by monitoring one of them and it cannot increase anymore if another remaining device is also monitored. This results in the nearly horizontal line from one to two devices in Figure 5.2(d). Besides, the performance gain cannot be improved too much when monitoring more other devices.



(a) REDD 1



(b) REDD 2

FIGURE 5.3: Performance improvement of the CPH algorithm when modifying WSN precision.

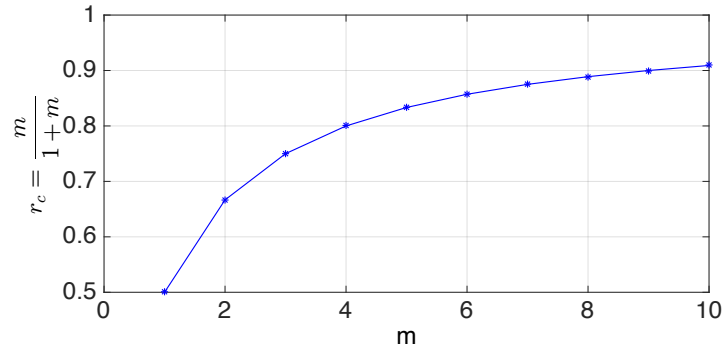


FIGURE 5.4: Dependence of the abnormally changing point on the number of states.

5.2.3 Impact of WSN detection on the performance of the Knapsack approach

In the experiments above, the WSN is assumed with a detection precision of 0.9. However, in real condition, this value can vary depending on the quality of the sensors. Therefore, we will consider the effect of the sensor detection on the overall performance of SmartSense algorithms by tuning the precision from 0.6 to 0.9 with two monitored

devices (fridge and dish washer) for REDD 1 and REDD 2. Figure 5.3 shows the performance gain of the CPH algorithm according to the increase of the WSN precision. In Figure 5.3(a), with monitored devices, the performance of the algorithm in REDD 1 decreases significantly if the precision of the sensors is lower than 0.66. This is illustrated by the negative gain of both precision and recall. Meanwhile, when increasing this value from 0.66 to 0.68, the performance abnormally increases with the precision gain from 9% to 16.5% and the recall from 1% to 9%. After this interval, the performance is slowly improved to 20% of precision gain and 15% of recall gain at a WSN precision of 0.9. Similarly, with REDD 2, as shown in Figure 5.3(b), the performance gain is also negative with the WSN precision less than 0.6 and there also exists an abnormally changing interval between 0.66 and 0.68, in which the precision gain increases from -7.5% to 12.5% and the recall from -12% to 8%. Beyond this interval, the performance slowly increases along with the WSN precision. To find the reason of the abnormally increasing interval, let return to Section 4.2. If an m -state device is detected as switched on, the corresponding probability of each power state is equal to pr/m and off-state is $(1 - pr)$. Thus, with $pr < m/(m + 1)$, the probability of the off-state is greater than any on-state although the device is operating and that makes the decrease of performance. In contrast, if $pr > m/(m + 1)$, the probability of any on-state is larger than off-state and the probability information allows to improve the detection accuracy. In both REDD 1 and REDD 2 dataset, the monitored devices are fridge and dish washer, each of which has two operation states, which creates a changing point at $pr \approx 0.67$. Obviously, with the edge detector approach, this phenomenon always happens at $pr = 0.5$. The dependence of the changing point $pr = \frac{m}{m+1}$ on the number of states of devices is presented in Figure 5.4.

As a consequence, the precision of the WSN needs to ensure to stay on the upper plane discriminated by the changing curve in order to allow the algorithm to reach a better performance.

5.2.4 Impact of monitored devices on the performance of the edge detector approach

Similar to CPH and DP algorithms, the performance of the DTW and ED algorithms in SmartSense also depends on the type of monitored devices. To analyze the main characteristics of the most effective devices, each device in REDD 1 and REDD 2 is sequentially

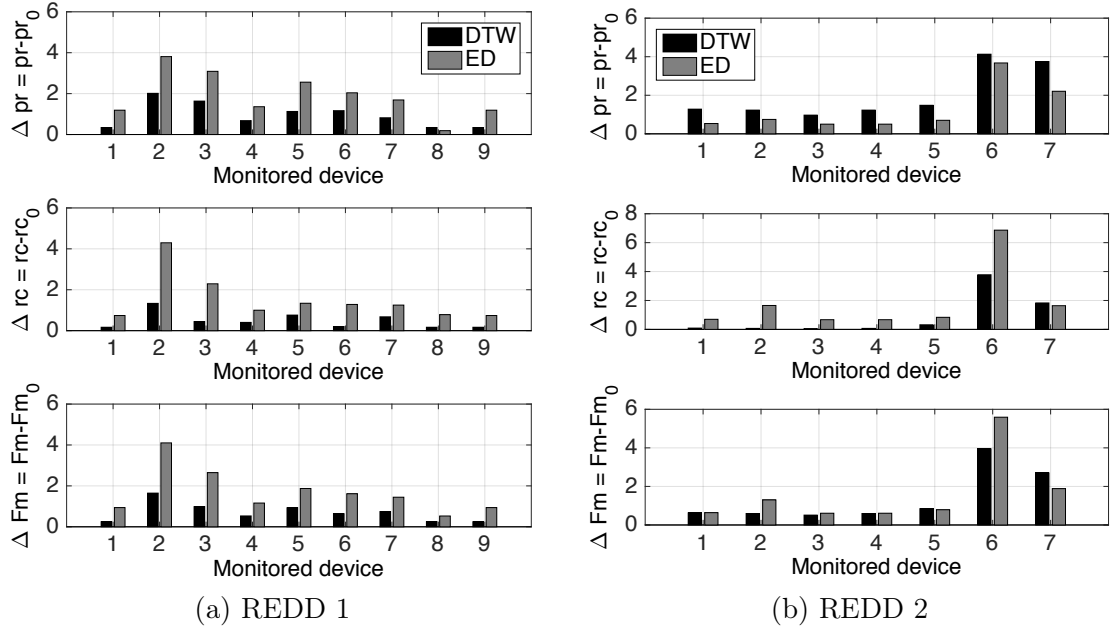


FIGURE 5.5: Effect of the type of monitored devices on the performance improvement of the DTW and ED algorithms. The values of pr_0 , rc_0 , Fm_0 correspond to the case without any monitored device, given in Table 5.6.

selected to monitored. The performance gain compared with the original algorithms in NILM, given in Table 5.6, is shown in Figure 5.5. Two devices in each dataset giving the best gain are listed in Table 5.12. As shown in Figure 5.5(a), the fridge is the best effective device to be monitored in REDD 1. By applying the ED algorithm, monitoring the fridge can achieve a precision gain of 4% and recall gain of 4.2%. Comparing with Table 5.2, because the fridge has the confusion on the power demand with the dish washer, both devices have a near value of edge height, which reduces the performance of the edge detector based algorithms in a traditional NILM system. By monitoring one of them, the ambiguity can be overcome. Besides, the fridge also has a high number of activations during the observation period (1695 times). Obviously, monitoring the devices with high number of uses allows to increase the overall performance of the algorithms. The second largest gain in REDD 1 corresponds to the dish washer with 3% and 2% for the gain of precision and recall, respectively. Besides confused with the fridge on the edges height, this device is also frequently switched on/off. Meanwhile, the performance improvement given by the DTW algorithm is lower than the ED one with only 2% of precision gain and 1.4% of recall gain when monitoring the fridge. The respective results for the dish washer are 1.9% and 0.25%. The small gain of the DTW algorithm with this dataset can be explained by the fact that it shows a good performance without any monitored device

TABLE 5.12: Most effective devices with DTW and ED algorithms.

| | DTW | ED |
|--------|---------------------------|---------------------------|
| REDD 1 | 2-fridge 3-dish washer | 2-fridge 3-dish washer |
| REDD 2 | 6-fridge 7-dish washer | 6-fridge 7-dish washer |

($pr = 83.53\%$, $rc = 76.48\%$) and it is difficult to improve its performance. In REDD 2, as shown in Figure 5.5(b), the best gain also corresponds to the fridge with 3.9% of precision and 6.9% of recall with ED algorithm. The respective values with DTW method are 4.05% and 3.9%. Similar to REDD 1, the fridge in REDD 2 is also the most frequently used device and its edges are also ambiguous with the lighting system (153W) and dish washer (426W) when it changes the state from 163W to 426W with a rising edge of 263W. This is also the reason why the dish washer is the second effective device in this dataset. From the results in Figure 5.5, it can be concluded that the edge detector based algorithms are sensible with the devices more frequently switched on/off and having the confusion on the edges height (correlated to the power demand) with the others. Therefore, it is necessary to analyze these two characteristics to select the effective monitored devices.

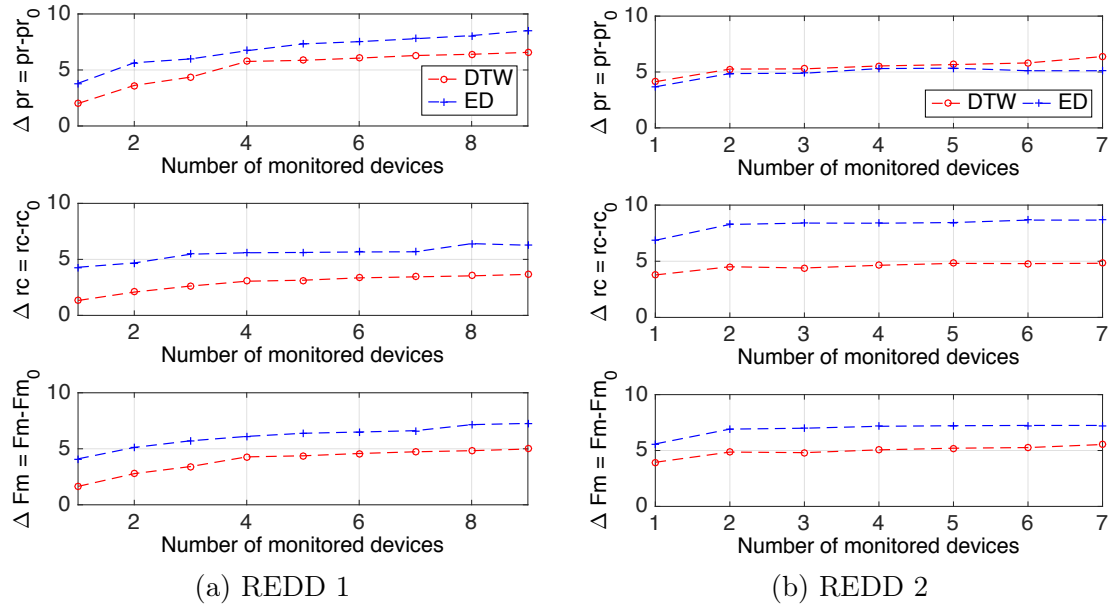
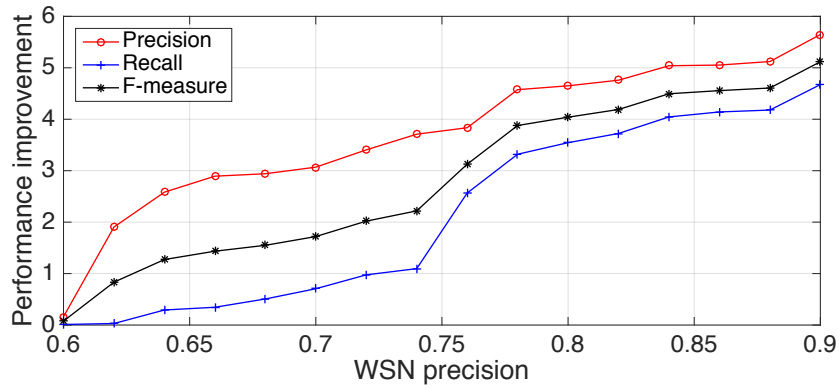
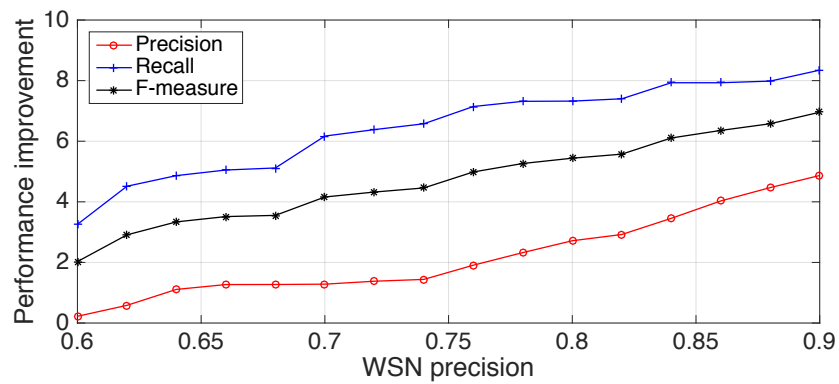


FIGURE 5.6: Effect of the number of monitored devices on the performance improvement of the DTW and ED algorithms. The values of pr_0 , rc_0 , Fm_0 correspond to the case without any monitored device, given in Table 5.6.

The performance gain also slowly increases when adding more devices to the monitoring list, as illustrated in Figure 5.6. In this results, the monitored devices are selected based on their performance in Figure 5.5. In Figure 5.6(a), with two monitored devices (fridge and dish washer) in REDD 1, the precision of the ED algorithm can be improved by 6% and the recall by 5%. Meanwhile, with other additional devices, the precision can only increase by 2.8% (from 6% to 8.8%) and the recall by 1.5% (from 5% to 6.5%). Similarly, in REDD 2 as shown in Figure 5.6(b), the performance gain obtainable with the ED algorithm is 5% of precision and 8.5% of recall with only fridge and dish washer monitored. Then, it increases very slowly when adding more devices to the monitoring list. Concretely, the precision gain increases from 5% to 6.5% and the recall from 8.5% to 9% with the number of devices from two to seven. Therefore, it is necessary to evaluate the efficiency of each device in improving the overall performance before deploying the monitoring sensors.



(a) REDD 1



(b) REDD 2

FIGURE 5.7: Performance improvement of the ED algorithm versus the WSN precision.

Similar to CPH and DP, the performance of the edge detector based algorithms also increases along with the increase of the WSN detection precision, as illustrated by the

simulation results of ED one with REDD 1 and REDD 2 in Figure 5.7. Because the operating probability of each device is not divided by the number of states, there is only one abnormally changing point at $pr = 0.5$ for these algorithms. This point is not presented in Figure 5.7 because the value of WSN precision is only tuned from 0.6 to 0.9. Moreover, a WSN precision of 0.5 is equivalent to a random selection of the state, which is not relevant in our case.

5.2.5 Performance comparison between the two approaches

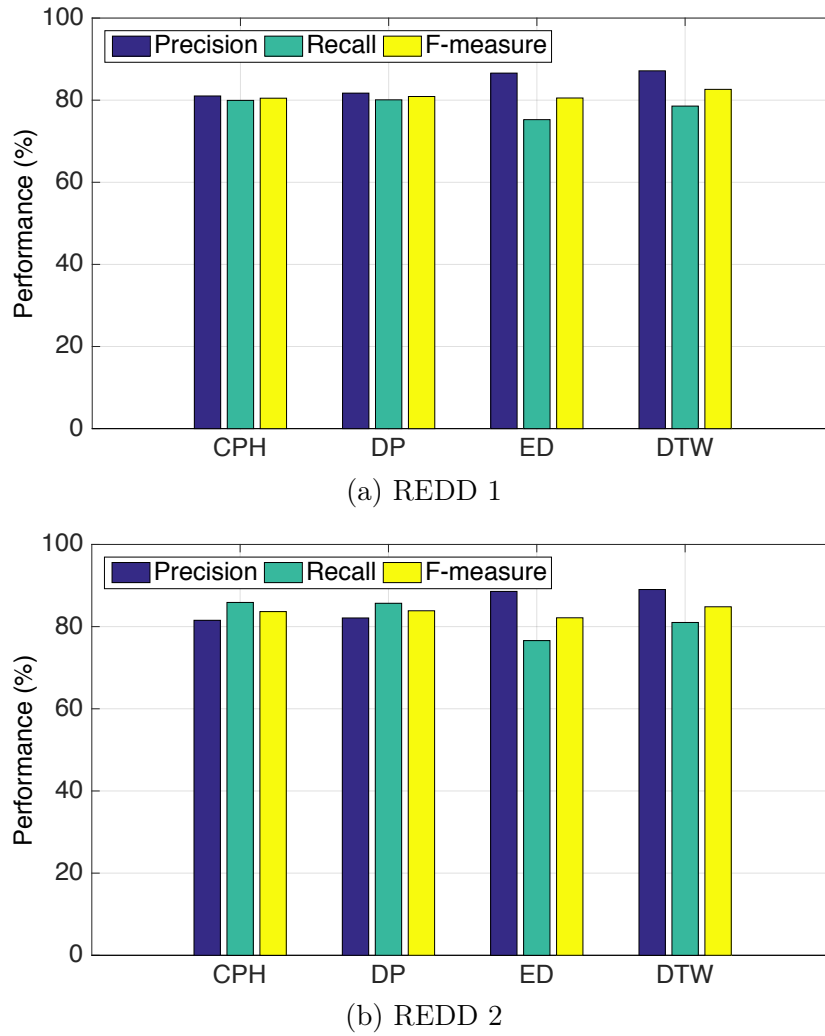


FIGURE 5.8: Performance comparison of the proposed algorithms with two monitored devices.

Comparing with the CPH and DP methods, it can be noticed that the improvement of the DTW and ED algorithms is much lower. As presented in Table 5.6, these algorithms show high performance without monitored device. Concretely, the F-measure of the ED

TABLE 5.13: Performance per device in REDD 1 (%).

| Device | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|--------|-----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| CPH | <i>pr</i> | 65.08 | 97.66 | 86.64 | 78.55 | 41.69 | 19.23 | 22.77 | 30.14 | 97.16 |
| | <i>rc</i> | 36.80 | 88.95 | 87.66 | 79.28 | 46.53 | 45.17 | 35.11 | 29.46 | 69.22 |
| | <i>Fm</i> | 47.01 | 93.10 | 87.15 | 78.91 | 43.97 | 26.98 | 27.63 | 29.79 | 80.84 |
| DP | <i>pr</i> | 63.94 | 99.16 | 96.50 | 76.55 | 43.79 | 22.50 | 20.63 | 19.91 | 94.68 |
| | <i>rc</i> | 35.84 | 91.59 | 92.13 | 76.00 | 41.20 | 46.26 | 42.84 | 25.71 | 73.34 |
| | <i>Fm</i> | 45.93 | 95.23 | 94.26 | 76.27 | 42.45 | 30.28 | 27.85 | 22.44 | 82.66 |
| ED | <i>pr</i> | 42.88 | 93.65 | 95.81 | 61.91 | 31.33 | 40.52 | 49.27 | 21.17 | 99.75 |
| | <i>rc</i> | 59.66 | 80.43 | 83.88 | 61.35 | 68.01 | 61.91 | 43.21 | 19.12 | 54.18 |
| | <i>Fm</i> | 49.89 | 86.54 | 89.45 | 61.63 | 42.90 | 48.98 | 46.04 | 20.09 | 70.72 |
| DTW | <i>pr</i> | 47.54 | 89.35 | 80.85 | 68.81 | 34.17 | 87.41 | 94.36 | 21.46 | 99.36 |
| | <i>rc</i> | 74.95 | 85.90 | 86.19 | 25.05 | 68.01 | 65.17 | 60.77 | 20.45 | 67.61 |
| | <i>Fm</i> | 58.18 | 87.59 | 83.43 | 36.73 | 45.48 | 74.67 | 73.93 | 20.94 | 40.47 |

algorithm is greater than 75% and the DTW one over 79%. Meanwhile, the CPH and DP algorithms can only show an F-measure lower than 70%. Obviously, high performance prevents the edge detector based algorithms from giving a high gain. In addition, the results also come from the fact that performance of these algorithms also depends on the edge detector capacity. Therefore, despite outperforming the algorithms with the knapsack approach in a traditional NILM system, both DTW and ED algorithms can only show a near performance in term of F-measure with two monitored devices, as illustrated in Figure 5.8. Of course, comparing the increase of the gain in Figures 5.2 and 5.6, the CPH and DP algorithms can show a better performance than the ED and DTW ones with more than two monitored devices in home. Besides the overall performance, the performance per device of each proposed algorithms with two monitored devices for each dataset (selected by the results in Figure 5.1 for CPH and DP algorithms and Figure 5.5 for ED and DTW ones) is also presented in Tables 5.13, 5.14, 5.15 and 5.16, respectively.

Apparently, the detection of the monitored devices is significantly improved. For example in REDD 1, as in Table 5.13, both fridge (2) and dish washer (3) are identified with the precision and recall greater than 85% by the CPH and DP algorithms and over 80% by the edge detector ones. Concretely, without sensors, the performance of the CPH algorithm with the fridge is only 62.13% of precision and 57.78% of recall as shown in Table 5.7, but they are significantly improved up to 97.66% and 88.95%, respectively, in SmartSense. Similarly, the precision of detecting the dish washer also increases from 23.83% to 81.64% and the recall increases from 69.28% to 87.66%. Meanwhile, the precision of the DTW

TABLE 5.14: Performance per device in REDD 2 (%).

| Device | | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|--------|-----------|-------|-------|-------|-------|-------|-------|-------|
| CPH | <i>pr</i> | 23.10 | 91.60 | 27.17 | 99.25 | 62.23 | 99.92 | 27.42 |
| | <i>rc</i> | 24.20 | 86.62 | 85.80 | 30.73 | 56.47 | 92.86 | 55.58 |
| | <i>Fm</i> | 23.64 | 89.04 | 41.27 | 46.94 | 59.21 | 96.26 | 36.72 |
| DP | <i>pr</i> | 27.18 | 95.53 | 26.02 | 97.33 | 48.57 | 99.95 | 26.82 |
| | <i>rc</i> | 43.19 | 90.49 | 86.80 | 21.84 | 38.09 | 88.44 | 57.10 |
| | <i>Fm</i> | 33.36 | 92.94 | 40.04 | 35.67 | 42.70 | 93.85 | 36.50 |
| ED | <i>pr</i> | 80.65 | 94.05 | 21.55 | 100 | 63.09 | 85.60 | 82.71 |
| | <i>rc</i> | 19.65 | 41.75 | 99.13 | 17.62 | 85.12 | 85.96 | 84.43 |
| | <i>Fm</i> | 31.60 | 57.82 | 35.40 | 29.96 | 72.47 | 85.78 | 83.56 |
| DTW | <i>pr</i> | 81.56 | 95.07 | 21.55 | 52.71 | 63.09 | 84.37 | 80.54 |
| | <i>rc</i> | 29.46 | 43.19 | 93.13 | 39.64 | 85.12 | 90.27 | 83.92 |
| | <i>Fm</i> | 43.29 | 59.40 | 35.01 | 45.62 | 72.47 | 87.22 | 82.19 |

TABLE 5.15: Performance per device in UK-DALE 5 (%).

| Device | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|--------|-----------|-------|-------|-------|-------|-------|-------|-------|-------|
| CPH | <i>pr</i> | 22.83 | 36.99 | 91.24 | 40.35 | 17.48 | 22.15 | 97.46 | 87.10 |
| | <i>rc</i> | 69.10 | 55.31 | 89.88 | 24.74 | 52.20 | 67.68 | 85.72 | 75.84 |
| | <i>Fm</i> | 34.32 | 44.33 | 90.55 | 30.67 | 26.19 | 33.38 | 91.21 | 81.08 |
| DP | <i>pr</i> | 23.57 | 40.77 | 90.57 | 37.47 | 25.81 | 23.08 | 98.31 | 88.52 |
| | <i>rc</i> | 58.02 | 80.19 | 89.15 | 22.66 | 53.18 | 72.60 | 87.45 | 65.03 |
| | <i>Fm</i> | 33.52 | 54.08 | 89.86 | 28.24 | 34.75 | 35.02 | 92.56 | 74.98 |

TABLE 5.16: Performance per device in Athemium dataset (%).

| Device | | 1 | 2 | 3 | 4 | 5 | 6 |
|--------|-----------|-------|-------|-------|-------|-------|-------|
| CPH | <i>pr</i> | 93.55 | 86.17 | 98.91 | 65.31 | 85.73 | 99.16 |
| | <i>rc</i> | 99.14 | 99.16 | 81.69 | 80.41 | 98.61 | 95.01 |
| | <i>Fm</i> | 96.27 | 92.21 | 89.48 | 72.08 | 91.72 | 97.04 |
| DP | <i>pr</i> | 86.43 | 89.73 | 100 | 65.60 | 82.63 | 99.41 |
| | <i>rc</i> | 99.46 | 99.16 | 91.09 | 87.80 | 98.89 | 87.92 |
| | <i>Fm</i> | 92.49 | 94.21 | 95.34 | 75.09 | 90.03 | 93.31 |

TABLE 5.17: Number of computations.

| | REDD 1 | | REDD 2 | |
|-----|--------------------|--------------------|--------------------|--------------------|
| | Multiplications | Additions | Multiplications | Additions |
| CPH | 3.36×10^5 | 3.64×10^7 | 1.10×10^5 | 9.20×10^6 |
| DP | 1.27×10^5 | 3.69×10^7 | 6.69×10^4 | 3.60×10^6 |
| ED | 3.14×10^3 | 1.68×10^5 | 2.45×10^3 | 1.05×10^5 |
| DTW | 4.60×10^3 | 1.47×10^7 | 3.71×10^3 | 3.80×10^6 |
| | UK-DALE 5 | | Athemium | |
| | Multiplications | Additions | Multiplications | Additions |
| CPH | 7.93×10^6 | 9.21×10^8 | 1.02×10^5 | 6.53×10^6 |
| DP | 1.55×10^6 | 3.58×10^8 | 7.62×10^4 | 2.94×10^6 |

approach increases from 85.39% to 89.35% when detecting the fridge and from 78.85% to 80.85% when detecting the dish washer. The respective recall is from 83.49% to 85.9% and from 56.19% to 86.19%. The performance improvement of the monitored devices in other dataset is similar. In SmartSense, not only the detection of the monitored devices but the identification of other loads is also improved. It comes from the fact that in each dataset, these loads have confusions on the power demand with the monitored ones and the additional information in SmartSense helps to overcome this ambiguity. For example in REDD 2, referring to Tables 5.8 and 5.14, the performance of the DTW algorithm when detecting the lighting system can be improved from 94.17% to 95.07% with precision and from 39.01% to 43.19% with recall because the fridge, which has one power state ambiguous with the lighting, is monitored by the sensors. Therefore, by monitoring only a subset of all devices, the overall performance of the NILM algorithms can be improved using the SmartSense approach.

5.2.6 Complexity analysis

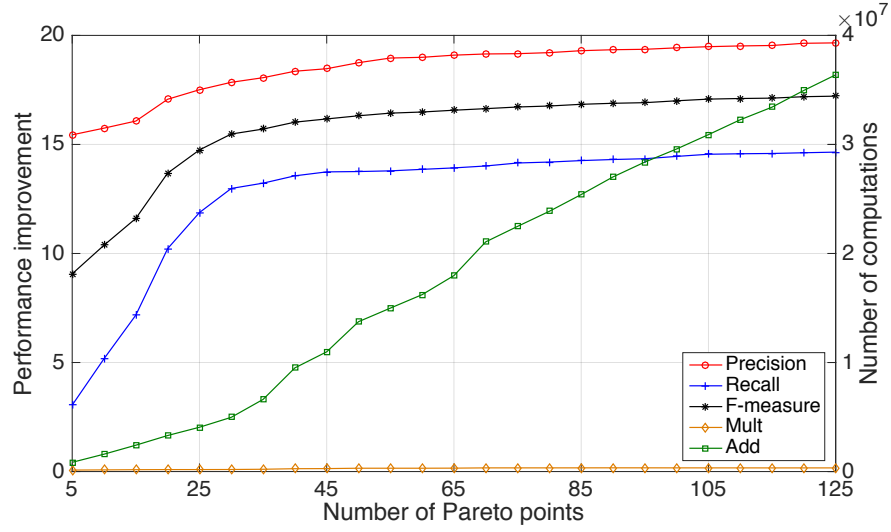
Algorithmic complexity, expressed through the number of multiplications and additions to process each dataset, is given in Table 5.17 for the proposed algorithms. These parameters do not depend on the number of monitored devices. Apparently, the CPH algorithm is the most complex because it requires a large number of multiplications and additions to calculate the elements of each tuple. From Section 4.2.2.1, the CPH algorithm has an exponential complexity in the worst case. If each device has m states and if all possible combinations are Pareto points, there are m^N tuples after the last iteration and the

Pareto minimization process has the complexity of m^{2N} . Nevertheless, if only L solutions are retained instead of all Pareto points at each iteration, there are only L^2 tuples appearing after combining two input sets. Hence, an L^4 -complex process is necessary to search for the Pareto points and that results in a complexity of NL^4 for the CPH algorithm. The selection of L best tuples can be executed as in [3]. Denote $r = d_1$ and $v = -d_3$, the Pareto points satisfy the following characteristic:

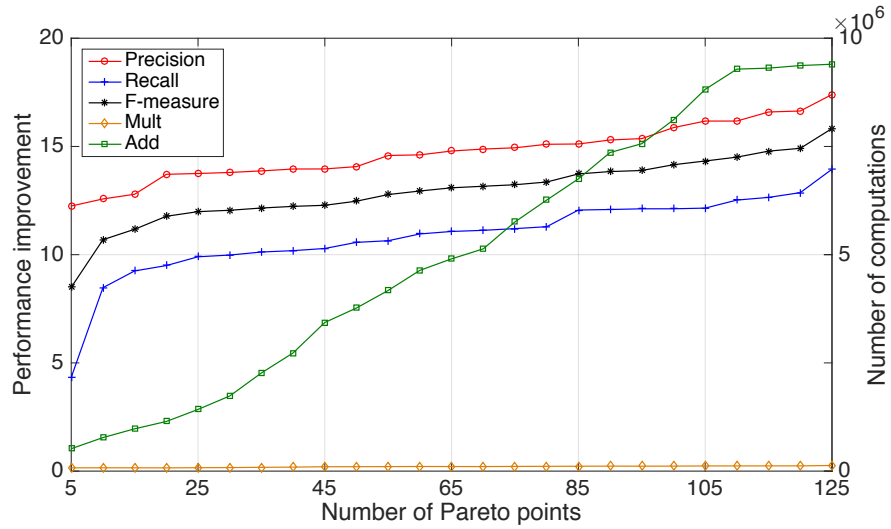
$$\forall i, j : v_i \leq v_j \leftrightarrow r_i \leq r_j. \quad (5.4)$$

At first, two Pareto points (r_{\max}, v_{\max}) and (r_{\min}, v_{\min}) are selected, which results in two constant product values $c_{\max} = v_{\max} \times r_{\max}$ and $c_{\min} = v_{\min} \times r_{\min}$. To select the $(L - 2)$ remaining points, the range $c_{\min}..c_{\max}$ will be divided into $(L - 1)$ parts equally discriminated by $(L - 2)$ boundary points c . For each of $(L - 2)$ values c , a Pareto point giving the product of $v \times r$ closest to c will be kept. The decrease of L allows to reduce the computational complexity. However, this increases the risk of rejecting the true combination from the final set of tuples and the decrease of the load disaggregation performance, as illustrated in Figure 5.9. In Figure 5.9(a), with two monitored devices in REDD 1, the performance gain can be obtained with 20% of precision and 14.5% of recall if there are 125 partial solutions retained after each iteration. This selection of L leads to a large number of multiplications (3.36×10^5) and additions (3.64×10^7) to execute the dataset. By reducing the value of L , the number of computations can significantly decrease, e.g. 1.81×10^5 of multiplications and 4.06×10^6 of additions with $L = 25$. However, the performance of the CPH algorithm also simultaneously decreases down to 17.5% of precision gain and 12% of recall gain. If keeping only 5 partial solutions, i.e. $L = 5$, the number of multiplications and additions decreases to 1.37×10^5 and 8.49×10^5 , respectively. Nevertheless, the corresponding obtainable precision and recall gain are only 15.5% and 3%. Similarly, in REDD 2 as shown in Figure 5.9(b), the number of multiplications and additions reduce from 1.10×10^5 and 9.20×10^6 to 7.55×10^4 and 5.18×10^5 , respectively, when changing the value of L from 125 to 5. The performance gain also decreases from 17.5% to 12.22% and 14% to 4.5% for the precision and recall, respectively. Therefore, driving the value of L rises the need to make a trade-off between the complexity and the desired performance.

Similar to CPH method, the DP algorithm also needs a large amount of computations.



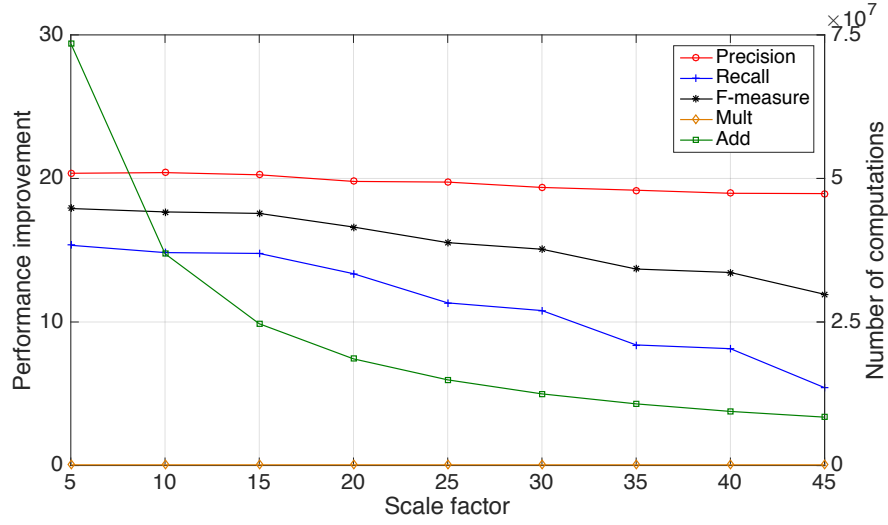
(a) REDD 1



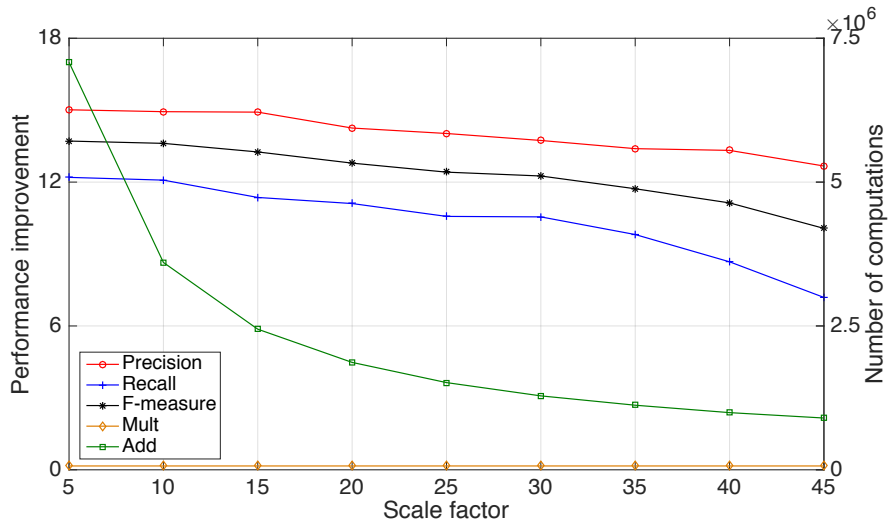
(a) REDD 2

FIGURE 5.9: Effect of the number of Pareto points on the performance and complexity of the CPH algorithm.

The principle of DP is based on the construction of profit tables. Therefore, the computational complexity depends on the size of each table, correlating to the number of devices, the aggregate power and the number of computations in each cell, decided by the number of states of current device. As a consequence, the complexity reduction effort can be executed by decreasing the number of columns with a scale factor δ , e.g. $w_{ij} = w_{ij}/\delta, i = 1, \dots, N, j = 1, \dots, m_i, R = R/\delta$. If possible, the best choice of δ is the common divisor of R and w_{ij} . However, this condition is unrealistic and the effect of the rounding down operation makes the performance decrease, as shown in Figure 5.10. In the DP algorithm, the construction of the profit tables is only based on the addition.



(a) REDD 1



(b) REDD 2

FIGURE 5.10: Effect of the scale factor on the performance and complexity of the DP algorithm.

This is the reason why the scale factor only affects the number of additions and does not change the number of multiplications, as illustrated in Figure 5.10(a). The results with REDD 1 in case with two monitored devices show a decrease of the number of additions from 7.35×10^7 to 8.43×10^6 when increasing the scale factor from 5 to 45. The performance of the algorithm also attenuates respectively by the precision gain from 24.5% to 19% and the recall gain from 15.2% to 5.5%. Meanwhile, Figure 5.10(b) presents the attenuation of the performance gain from 15% to 12.7% of precision and from 12.2% to 7.2% of recall when the value of δ increases from 5 to 45 in REDD 2. This tuning also allows to significantly decrease the number of computations. Concretely, the additions

reduce from 7.08×10^6 to 9×10^5 .

Different from the algorithms in the knapsack approach, both ED and DTW are less complex. Apparently, the computational complexity of the ED algorithm depends on the number of patterns in the library. If each of the N devices has m patterns, the device identification procedure needs to browse Nm times to find the minimum distance. Meanwhile, because taking all power values in a pattern into account, besides the number of patterns in the library, the complexity of the DTW algorithm is also decided by the length of each one. If each pattern has an average length of l , the accumulated distance calculation has a complexity of l^2 , which makes the DTW become an Nml^2 -complex algorithm. Despite less complexity, both ED and DTW algorithms are not suitable for the real-time application. They need to detect the falling edge before pairing with the rising one and identifying the corresponding device during this period. On the contrary, when applied to real-time energy disaggregation, the complexity of the CPH and DP one is compatible for implementation in an NILM system, since the algorithm is applied on the main system measuring the global energy consumption.

5.3 Conclusions

In this chapter, we introduce a new load monitoring system, which uses a WSN to detect and provide the operating probability of some devices to help the NILM algorithms in improving their performance. This system is called sensor-aided non-intrusive load monitoring (SmartSense). Although deploying a sensors network, SmartSense is less intrusive than other intrusive load monitoring systems such as ViridiScope [6], because only a subset of all devices is selected to attach with the sensors and the identification is still based on an NILM algorithm. The operation probability is used as a feature and combined with the traditional features of NILM by a regularization parameter to create a new one. In this chapter, two approaches are proposed:

- Knapsack approach: the l_1 -norm minimization problem in [1] is modeled as a knapsack problem and solved by two proposed algorithms including compositional Pareto-algebraic heuristic, which is based on a recursive relation, and dynamic programming, based the the construction of profit tables.

- Edge detector approach: an edge detector is applied to detect the rising edge and falling edge of an activation on the aggregate power signal and to compare them with the existing ones in the library. The rising edge and falling edge can be directly used as a feature (as in edge detection algorithm) or combined with other active power values between them (as in dynamic time warping algorithm).

By simulating all four proposed algorithms with four dataset including UK-DALE 5, REDD 1, REDD 2 and our Athemium data, the performance of the load detection can be significantly improved with some monitored devices in comparison with traditional NILM systems. To obtain a good improvement, the devices selected to be monitored need to satisfy some criteria. In the knapsack approach, selected devices are the one with high using rate and confusion on the power demand with other devices. Meanwhile, the edge detector approach needs to monitor the devices more frequently switched on/off and ambiguous with the others on the edges height. As a consequence, only several devices in home or building are selected, satisfying these requirements and allowing to significantly improve the overall performance when monitored. The others are less effective and need to be ignored when deploying the monitoring sensor network to reduce the cost. In a practical situation, we have an existing installation to monitor, and all we have is limited information about it. Users can therefore choose *a priori* some loads to be monitored based on their anticipation on the using frequency of each device.

Besides, comparing the performance of the proposed approaches shows that the CPH and DP algorithms outperform the ED and DTW ones with more than two monitored devices in each dataset, although in normal NILM system, they are less effective. This result comes from the fact that the performance of the ED and DTW algorithms also depends on the edge detector capacity.

However, the algorithms of the knapsack approach show a high complexity presented by the large amount of multiplications and additions. To reduce the computational complexity, we can drive some parameters such as the number of Pareto points retained after each iteration in CPH or the scale factor in DP. Nevertheless, it also causes the attenuation of the performance.

Chapter 6

Conclusions and Perspectives

6.1 Conclusions

In this thesis, we considered state detection and load disaggregation, which can be implemented by three main approaches:

- Intrusive load monitoring: the sensors (including the power meters and the environment monitoring sensors) are attached to each individual device. This approach obviously shows a high accuracy, but requires a high cost as well as too much technical intervention on the power supply to deploy the sensors.
- Non-intrusive load monitoring: this approach is promising to study because it requires only one power meter installed at the main power entry to measure the power consumption of all devices. The aggregate power signal is then analyzed to extract the specific features to identify the operation of the corresponding devices. The selection of features depends on the sampling frequency of the meters, divided into two groups:
 - High frequency: the features relate to the transient signal waveform, e.g. *v-shape*, or the Fourier transform of the stable signal, e.g. harmonics.
 - Low frequency: basically corresponding to the power signal of the steady state such as average power, step-changes of length of each steady duration.

Apparently, high-frequency based methods can benefit from more characteristics of the devices and show a higher quality of detection than the low frequency approaches. However, the price of the hardware also increases along with the frequency and the identification is more complex. For example, it is necessary to transform the signal before applying the algorithms.

- Hybrid load monitoring: to improve the performance of the NILM algorithms, some additional information such as acoustic noise, light intensity, occupancy can be used to detect the operation of some specific devices. Obviously, this approach leads to a high cost than NILM, but it is still less intrusive.

In our work, in Chapter 3, we firstly try to solve the $l1$ -norm minimization problem in NILM by a brute force algorithm. This method sequentially tests all combinations of states to find one giving the minimum absolute error between the aggregate power and total power demand of the identified devices. Obviously, this algorithm cannot show a good performance if there are devices with the same or near power demand. Therefore, instead of selecting the combination with the least error, we proposed to keep all combinations giving an error around the least value and compare with the previous state of the devices to determine the final solution. The comparison can be based on the distance from the previous state and the current state or the state transition probability between them. The simulation results with the Athemium data, retrieved by monitoring some devices in our coffee room, show that the performance can be significantly improved by these two versions of the brute force algorithm.

The main contribution of this thesis is presented in Chapter 4 with a hybrid load monitoring system called SmartSense, in which a WSN is deployed to monitor some specific devices and provide their operating probability. There are two approaches proposed to combine the probability feature with the NILM algorithms, including:

- Knapsack approach: formulated by combining the probability of the devices with the minimization problem in NILM. The Knapsack problem is then solved by a compositional Pareto-algebraic heuristic (CPH) or a dynamic programming (DP) algorithm.
- Edge detector approach: an edge detector is applied to detect the rising edges and falling edges of an activation on the power signal to create the features. The

probability information of each device is then combined with these features by a regularization parameter to form new more effective ones. If the ED algorithm is applied, the features comprise only the rising edge and falling edge, while in the DTW one, all active power values between these edges are retained.

In Chapter 5, the proposed algorithms are simulated with four dataset including two houses from REDD, one house in UK-DALE and our Athemium data. The results show that the performance can be significantly improved by monitoring a part of devices. Besides, there are several parameters affecting the performance of the algorithms in SmartSense, such as type and number of monitored devices or the precision of the sensor network. The selection of these parameters needs to make balance between the desired performance and the cost of WSN deployment. Besides, the computational complexity is also necessary to be considered. Despite performing a higher performance with more than two monitored devices, the CPH and DP algorithms require a larger amount of computations than the ED and DTW ones.

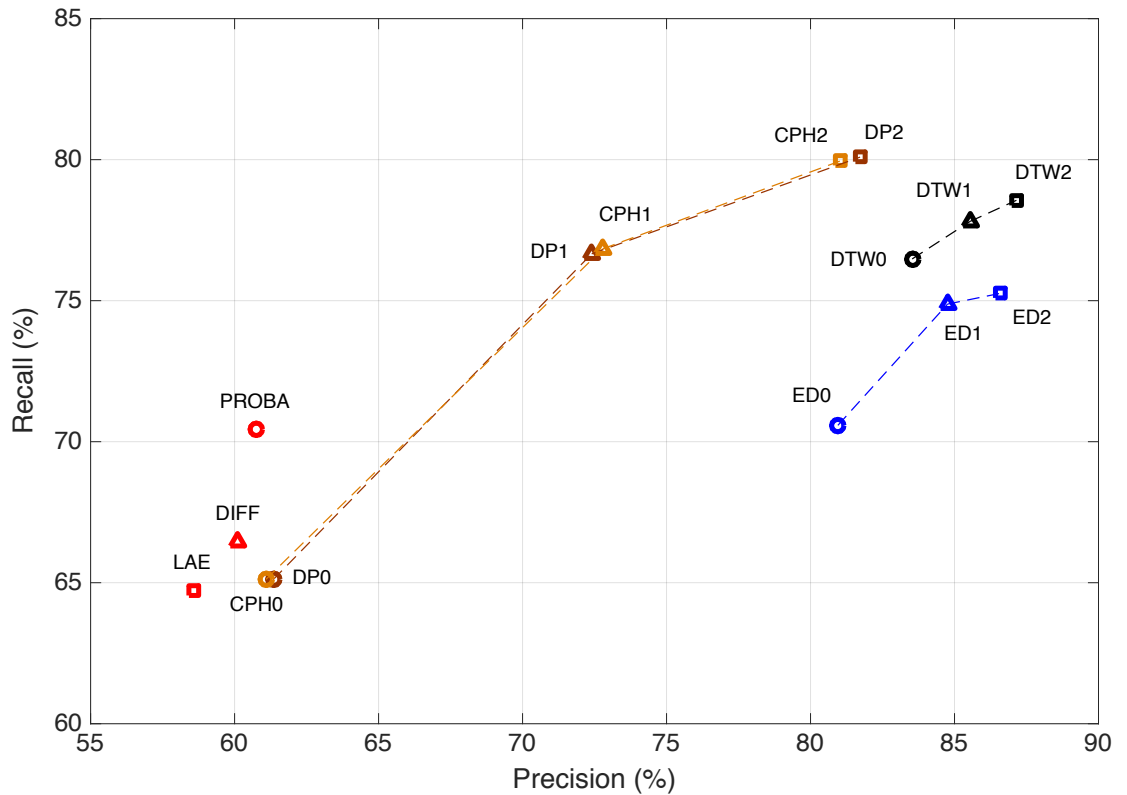


FIGURE 6.1: Performance of all proposed algorithms with REDD dataset *House 1*. The index following the algorithms denotes the number of monitored devices.

In Figure 6.1, performance of all proposed algorithms in this thesis with REDD dataset *House 1* is presented. The least absolute error based algorithm (LAE), state difference

based algorithm (DIFF) and state transition probability based algorithm (PROBA) are introduced in Chapter 3. Meanwhile, the four proposed algorithms in SmartSense with one monitored device (DP1, CPH1, ED1, DTW1) and two monitored devices (DP2, CPH2, ED2, DTW2) are compared with the corresponding algorithms in state of the art (DP0, CPH0, ED0, DTW0). The three l_1 -norm minimization based algorithms introduced in Chapter 3 (LAE, DIFF, PROBA) show the worst detection. Meanwhile, in the SmartSense system, although CPH and DP show a worse performance when applied in traditional NILM, the algorithms used in the knapsack approach are more significantly improved than the edge detector algorithms (DTW1, DTW2, ED1, ED2) when several devices in home are monitored by the WSN (DP1, DP2, CPH1, CPH2). In addition, they also outperform other algorithms in term of recall with more than one monitored devices.

6.2 Perspectives

Along with the development of smart homes and smart buildings, load monitoring becomes one of the most important issues, in which non-intrusive approach is the more promising because of less technical intervention on the power supply. Sensor-aided NILM or SmartSense, can become an efficient model for NILM researches because it can significantly improve performance of existing NILM algorithms with the deployment of several low-cost, low-power and less complicated sensors to monitor a subset of all devices in home or building. For example, the operation of the fridge compressor can be easily detected by a vibration sensor, while the lighting system can be monitored by a light sensor. These sensors do not require too much technical intervention and the users can easily install the sensors by themselves. As a consequence, the intrusive requirement is still respected, which is quite different from the intrusive load monitoring such as ViridiScope [6]. In addition, because the detection of sensors is used to estimate the operating probability of corresponding devices and used as an additional information in NILM algorithms, effect of false detection can be reduced.

Using sensors to monitor the operation of electrical devices has been developed for many decades. Moreover, the appearance of the electrical devices with embedded sensors such as smart televisions, smart lamps, etc., allows the SmartSense model to be deployed more conveniently. The opportunities for extending the scope of this thesis correspond

to the development of disaggregation algorithms as well as the improvement of probability estimation.

The most significant advantage of the SmartSense model is the reuse of existing NILM algorithms. In this thesis, we applied the SmartSense idea to the edge detector algorithms, which is the first approach of NILM proposed by G. Hart [1], and the proposed knapsack approach to solve the $l1$ -norm minimization problem in NILM. The SmartSense model shows results with significant improvements thanks to the different proposed algorithms. Of course, other approaches can also be applied in SmartSense such as HMM, ANN, deep learning neural networks, etc. Concretely, in any HMM, emission probability and transition probability are two of the most important parameters to be learned. When applied in SmartSense, the emission probability corresponding to each state can be studied to be replaced by the operating probability estimated by the WSN. Meanwhile, with ANN or deep learning neural network, the operating probability of each device can be considered as an input, along with the aggregate power measurement, to determine the state of devices at the output. The number of hidden layers and the structure of the network are principal issues to be studied in future work. Besides, we also need to study self-training algorithms, which are able to automatically learn the parameters by an adaptive process without training data. Algorithm which can be applied to the devices with variable loads is also an important issue.

In the knapsack approach of this thesis, the operating probability of each state strongly depends on the number of states of each device. This becomes a problem if a device has too many operating states. Because of being divided by the number of states, the operating probability of each state becomes very low although the corresponding device is detected as being switched on with high precision. Therefore, developing an algorithm for probability estimation in order to eliminate the effect of the number of states also needs to be studied in future work. Besides, the operating probability of cycling loads such as fridge can also be reported from the history of data series. This allows to increase the precision of the detection. Additionally, we also need to deploy a WSN in certain rooms to verify the operation of SmartSense in real condition instead of simulating with publicly dataset. This will be done during 2017 in the context of the SmartSense platform deployment funded by INRIA and the "Region Bretagne" in the context of the CPER SmartSense funding.

Personal Publications

- **X.-C. Le**, B. Vrigneau, O. Sentieys. l1-norm Minimization based Algorithm for Non-Intrusive Load Monitoring. *IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), IEEE Workshop on Pervasive Energy Services*, Mar 2015, St. Louis, United States. pp.299 - 304.
- **X.-C. Le**, B. Vrigneau, O. Sentieys. Operating Probability in Improving Performance of Non-Intrusive Load Monitoring Algorithms. *Submitted on IEEE Transactions on Smart Grid*.
- **X.-C. Le**, B. Vrigneau, O. Sentieys. Dynamic Programming in Sensor-aided Non-Intrusive Load Monitoring. *Submitted on ACM Transactions on Sensor Networks*.

Bibliography

- [1] G.W. Hart. Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 80 (12):1870–1891, Dec 1992. ISSN 0018-9219. doi: 10.1109/5.192069.
- [2] Jing Liao, Georgia Elafoudi, Lina Stankovic, and Vladimir Stankovic. Non-intrusive appliance load monitoring using low-resolution smart meter data. In *In Proceedings of the 5th Annual IEEE International Conference on Smart Grid Communications*, pages 541–546, Venice, Italy, 2014.
- [3] H. Shojaei, T. Basten, M. Geilen, and A. Davoodi. A fast and scalable multidimensional multiple-choice knapsack heuristic. *ACM Transactions on Design Automation of Electronic Systems*, 18(4), Oct. 2013.
- [4] J. Zico Kolter and Matthew J. Johnson. REDD: A public data set for energy disaggregation research. In *in SustKDD*, 2011.
- [5] Dong Chen, Sean Barker, Adarsh Subbaswamy, David Irwin, and Prashant Shenoy. Non-intrusive occupancy monitoring using smart meters. In *Proceedings of the 5th ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, pages 1–8, New York, NY, USA, 2013.
- [6] Younghun Kim, Thomas Schmid, Zainul M. Charbiwala, and Mani B. Srivastava. Viridiscopes: Design and implementation of a fine grained power monitoring system for homes. In *Proc. Ubicomp09*, pages 245–254, 2009.
- [7] Ahmed Zoha, Alexander Gluhak, Muhammad Ali Imran, and Sutharshan Rajasegarar. Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey. *Sensors*, 12(12):16838, 2012. ISSN 1424-8220. doi: 10.3390/s121216838. URL <http://www.mdpi.com/1424-8220/12/12/16838>.

- [8] Anthony Rowe, Mario Berges, and Raj Rajkumar. Contactless sensing of appliance state transitions through variations in electromagnetic fields. In *Proceedings of the 2Nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, BuildSys '10, pages 19–24, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0458-0. doi: 10.1145/1878431.1878437. URL <http://doi.acm.org/10.1145/1878431.1878437>.
- [9] Deokwoo Jung and Andreas Savvides. Estimating building consumption breakdowns using on/off state sensing and incremental sub-meter deployment. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 225–238, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0344-6. doi: 10.1145/1869983.1870006. URL <http://doi.acm.org/10.1145/1869983.1870006>.
- [10] M. Hazas, A. Friday, and J. Scott. Look back before leaping forward: Four decades of domestic energy inquiry. *IEEE Pervasive Computing*, 10(1):13–19, Jan. 2011. ISSN 1536-1268. doi: 10.1109/MPRV.2010.89.
- [11] A. Prudenzi. A neuron nets based procedure for identifying domestic appliances pattern-of-use from energy recordings at meter panel. In *Power Engineering Society Winter Meeting, 2002. IEEE*, volume 2, pages 941–946 vol.2, 2002. doi: 10.1109/PESW.2002.985144.
- [12] Jack Kelly and William Knottenbelt. Neural nilm: Deep neural networks applied to energy disaggregation. In *Proceedings of the 2Nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*, BuildSys'15, pages 55–64, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3981-0. doi: 10.1145/2821650.2821672. URL <http://doi.acm.org/10.1145/2821650.2821672>.
- [13] Oliver Parson, Siddhartha Ghosh, Mark Weal, and Alex Rogers. Non-intrusive load monitoring using prior models of general appliance types. In *1st International Workshop on Non-Intrusive Load Monitoring*, pages 356–362, Pittsburgh, PA, USA, 2012.
- [14] Hyungsul Kim, Manish Marwah, Martin Arlitt, Geoff Lyon, and Jiawei Han. Unsupervised disaggregation of low frequency power measurements. In *11th International Conference on Data Mining*, pages 747–758, Arizona, USA, 2011.

- [15] C. Laughman, Kwangduk Lee, R. Cox, S. Shaw, S. Leeb, L. Norford, and P. Armstrong. Power signature analysis. *Power and Energy Magazine, IEEE*, 1(2):56–63, Mar 2003. ISSN 1540-7977. doi: 10.1109/MPAE.2003.1192027.
- [16] S.B. Leeb, S.R. Shaw, and Jr. Kirtley, J.L. Transient event detection in spectral envelope estimates for nonintrusive load monitoring. *Power Delivery, IEEE Transactions on*, 10(3):1200–1210, Jul 1995. ISSN 0885-8977. doi: 10.1109/61.400897.
- [17] D. Srinivasan, W. S. Ng, and A.C. Liew. Neural-network-based signature recognition for harmonic source identification. *Power Delivery, IEEE Transactions on*, 21(1):398–405, Jan 2006. ISSN 0885-8977. doi: 10.1109/TPWRD.2005.852370.
- [18] H. Y. Lam, G. S. K. Fung, and W. K. Lee. A novel method to construct taxonomy electrical appliances based on load signaturesof. *IEEE Transactions on Consumer Electronics*, 53(2):653–660, May 2007. ISSN 0098-3063. doi: 10.1109/TCE.2007.381742.
- [19] Sidhant Gupta, Matthew S. Reynolds, and Shwetak N. Patel. Electrisense: Single-point sensing using emi for electrical event detection and classification in the home. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, pages 139–148, Copenhagen, Denmark, 2010.
- [20] K. Y. Chen, S. Gupta, E. C. Larson, and S. Patel. Dose: Detecting user-driven operating states of electronic devices from a single sensing point. In *Pervasive Computing and Communications (PerCom), 2015 IEEE International Conference on*, pages 46–54, Mar. 2015. doi: 10.1109/PERCOM.2015.7146508.
- [21] Guoming Tang and Kui Wu. A framework for occupancy-aided energy disaggregation. In *Proceedings of the Seventh International Conference on Future Energy Systems, Waterloo, ON, Canada, June 21 - 24, 2016 - Poster Sessions*, pages 2:1–2:2, 2016. doi: 10.1145/2939912.2939914. URL <http://doi.acm.org/10.1145/2939912.2939914>.
- [22] Mario Berges, Lucio Soibelman, and H. Scott Matthews. Leveraging data from environmental sensors to enhance electrical load disaggregation algorithms. In *Proceedings of the 13th International Conference on Computing in Civil and Building Engineering*, Nottingham, UK, jun 2010. URL http://www.marioberges.com/pubs/2010_berges_icccbce.pdf.

- [23] <https://www.smartgrid.gov/>, .
- [24] S. Galli, A. Scaglione, and Zhifang Wang. Power line communications and the smart grid. In *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pages 303–308, Oct 2010. doi: 10.1109/SMARTGRID.2010.5622060.
- [25] S. Galli, A. Scaglione, and Zhifang Wang. For the grid and through the grid: The role of power line communications in the smart grid. *Proceedings of the IEEE*, 99(6):998–1027, June 2011. ISSN 0018-9219. doi: 10.1109/JPROC.2011.2109670.
- [26] V.C. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G.P. Hancke. Smart grid technologies: Communication technologies and standards. *Industrial Informatics, IEEE Transactions on*, 7(4):529–539, Nov 2011. ISSN 1551-3203. doi: 10.1109/TII.2011.2166794.
- [27] L.T. Berger, A. Schwager, and J.J. Escudero-Garzas. Power line communications for smart grid applications. *Journal of Electrical and Computer Engineering*, 2013, 2013. URL <http://dx.doi.org/10.1155/2013/712376>.
- [28] Yuvraj Agarwal, Rajesh Gupta, Thomas Weng, and Bharathan Balaji, Bharathan andx Dutta. Duty-cycling buildings aggressively: The next frontier in HVAC control. In *IEEE/ACM Information Processor in Sensor Networks*, pages 246–257, Chicago, IL, USA, April 2011.
- [29] T. Weng and Y. Agarwal. From buildings to smart buildings-sensing and actuation to improve energy efficiency. *Design Test of Computers, IEEE*, 29(4):36–44, 2012. ISSN 0740-7475. doi: 10.1109/MDT.2012.2211855.
- [30] Yuvraj Agarwal, Bharathan Balaji, Rajesh Gupta, Jacob Lyles, Michael Wei, and Thomas Weng. Occupancy-driven energy management for smart building automation. In *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, pages 1–6, New York, NY, USA, 2010.
- [31] Jiakang Lu, Tamim Sookoor, Vijay Srinivasan, Ge Gao, Brian Holben, John Stankovic, Eric Field, and Kamin Whitehouse. The smart thermostat: Using occupancy sensors to save energy in homes. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys ’10, pages 211–224, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0344-6. doi: 10.1145/1869983.1870005.

- [32] Shengwei Wang, John Burnett, and Hoishing Chong. Experimental validation of co2-based occupancy detection for demand-controlled ventilation. *Indoor and Built Environment*, 8(6):377–391, 1999. doi: 10.1177/1420326X9900800605. URL <http://ibe.sagepub.com/content/8/6/377.abstract>.
- [33] A. Ridi, C. Gisler, and J. Hennebert. User interaction event detection in the context of appliance monitoring. In *Pervasive Computing and Communication Workshops (PerCom Workshops), 2015 IEEE International Conference on*, pages 323–328, Mar. 2015. doi: 10.1109/PERCOMW.2015.7134056.
- [34] M. Zeifman and K. Roth. Nonintrusive appliance load monitoring: Review and outlook. *IEEE Transactions on Consumer Electronics*, 57(1):76–84, 2011. ISSN 0098-3063. doi: 10.1109/TCE.2011.5735484.
- [35] M. Baranski and Voss J. Nonintrusive appliance load monitoring based on an optical sensor. In *Power Tech Conference Proceedings, 2003 IEEE Bologna*, volume 4, pages 8 pp. Vol.4–, Jun. 2003. doi: 10.1109/PTC.2003.1304732.
- [36] Mario Berges and Anthony Rowe. Appliance classification and energy management using multi-modal sensing. In *Proceedings of the Third ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, BuildSys ’11, pages 51–52, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0749-9. doi: 10.1145/2434020.2434037. URL <http://doi.acm.org/10.1145/2434020.2434037>.
- [37] M. Amac Guvensan, Z. Cihan Taysi, and Tommaso Melodia. Energy monitoring in residential spaces with audio sensor nodes: TinyEARS. *Ad Hoc Networks*, 11(5):1539 – 1555, 2013. ISSN 1570-8705. doi: <http://dx.doi.org/10.1016/j.adhoc.2012.10.002>. URL <http://www.sciencedirect.com/science/article/pii/S1570870512001898>.
- [38] M. Uddin and T. Nadeem. Energysniffer: Home energy monitoring system using smart phones. In *2012 8th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 159–164, Aug. 2012.
- [39] Deokwoo Jung and Andreas Savvides. Theory and algorithm of estimating energy consumption breakdowns using on/off state sensing. *ACM Trans. Sen. Netw.*, 11(1):5:1–5:36, sep 2014. ISSN 1550-4859. doi: 10.1145/2630880. URL <http://doi.acm.org/10.1145/2630880>.

- [40] Christian Beckel, Wilhelm Kleiminger, Thorsten Staake, and Silvia Santini. Improving device-level electricity consumption breakdowns in private households using on/off events. *SIGBED Rev.*, 9(3):32–38, July 2012. ISSN 1551-3688. doi: 10.1145/2367580.2367586. URL <http://doi.acm.org/10.1145/2367580.2367586>.
- [41] <http://qualisteo.com/>.
- [42] <http://smart-impulse.com/>, .
- [43] Steven Drenker and Ab Kader. Nonintrusive monitoring of electric loads. *IEEE Computer Applications in Power*, 12(4):47–51, 1999.
- [44] Leslie K. Norford and Steven B. Leeb. Non-intrusive electrical load monitoring in commercial buildings based on steady-state and transient load-detection algorithms. *Energy and Buildings*, 24(1):51 – 64, 1996. ISSN 0378-7788. doi: [http://dx.doi.org/10.1016/0378-7788\(95\)00958-2](http://dx.doi.org/10.1016/0378-7788(95)00958-2).
- [45] M.L. Marceau and R. Zmeureanu. Nonintrusive load disaggregation computer program to estimate the energy consumption of major end uses in residential buildings. *Energy Conversion and Management*, 41(13):1389 – 1403, 2000. ISSN 0196-8904. doi: [http://dx.doi.org/10.1016/S0196-8904\(99\)00173-9](http://dx.doi.org/10.1016/S0196-8904(99)00173-9).
- [46] A.I. Cole and A. Albicki. Data extraction for effective non-intrusive identification of residential power loads. In *Instrumentation and Measurement Technology Conference, 1998. IMTC/98. Conference Proceedings. IEEE*, volume 2, pages 812–815 vol.2, May 1998. doi: 10.1109/IMTC.1998.676838.
- [47] C.-C. Chuang, T. J. T. Sung, G.-Y. Lin, J. Y. C. Wen, and R.-I Chang. Non-intrusive appliance monitoring now: Effective data, generative modelling and LETE. In *ENERGY 2011: The First International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies*, pages 81–86, Venice/Mestre, Italy, May 2011.
- [48] N. Batra, H. Dutta, and A. Singh. Indic: Improved non-intrusive load monitoring using load division and calibration. In *Machine Learning and Applications (ICMLA), 2013 12th International Conference on*, volume 1, pages 79–84, Dec 2013. doi: 10.1109/ICMLA.2013.21.

- [49] Linda Farinaccio and Radu Zmeureanu. Using a pattern recognition approach to disaggregate the total electricity consumption in a house into the major end-uses. *Energy and Buildings*, 30(3):245 – 259, 1999. ISSN 0378-7788. doi: [http://dx.doi.org/10.1016/S0378-7788\(99\)00007-9](http://dx.doi.org/10.1016/S0378-7788(99)00007-9). URL <http://www.sciencedirect.com/science/article/pii/S0378778899000079>.
- [50] J. Z. Kolter, Siddharth Batra, and Andrew Y. Ng. Energy disaggregation via discriminative sparse coding. In J.D. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1153–1161. Curran Associates, Inc., 2010. URL <http://papers.nips.cc/paper/4054-energy-disaggregation-via-discriminative-sparse-coding.pdf>.
- [51] N. Pathak, N. Roy, and A. Biswas. Iterative signal separation assisted energy disaggregation. In *Green Computing Conference and Sustainable Computing Conference (IGSC), 2015 Sixth International*, pages 1–8, Dec. 2015. doi: 10.1109/IGCC.2015.7393701.
- [52] A. G. Ruzzelli, C. Nicolas, A. Schoofs, and G. M. P. O’Hare. Real-time recognition and profiling of appliances through a single electricity sensor. In *2010 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 1–9, Jun. 2010. doi: 10.1109/SECON.2010.5508244.
- [53] H. Najmeddine, K. El Khamlichi Drissi, C. Pasquier, C. Faure, K. Kerroum, A. Diop, T. Jouannet, and M. Michou. State of art on load monitoring methods. In *Power and Energy Conference, 2008. PECon 2008. IEEE 2nd International*, pages 1256–1258, Dec. 2008. doi: 10.1109/PECON.2008.4762669.
- [54] B. Ribeiro M. Figueiredo, A. de Almeida. An experimental study on electrical signature identification of non-intrusive load monitoring (NILM) systems. *Adaptive and Natural Computing Algorithms*, pages 31–40. ISSN 0302-9743. doi: 10.1007/978-3-642-20267-4_4.
- [55] Jack Kelly and William Knottenbelt. The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes. *Scientific Data*, 2(150007), 2015. doi: 10.1038/sdata.2015.7.

- [56] N. Batra, J. Kelly, O. Parson, H. Dutta, W. Knottenbelt, A. Rogers, A. Singh, and M. Srivastava. NILMTK: An open source toolkit for non-intrusive load monitoring. *ArXiv e-prints*, Apr. 2014.
- [57] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL <http://dx.doi.org/10.1162/neco.1997.9.8.1735>.
- [58] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning, ICML '08*, pages 1096–1103, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. doi: 10.1145/1390156.1390294. URL <http://doi.acm.org/10.1145/1390156.1390294>.
- [59] S. N. Akshay Uttama Nambi, T. G. Papaioannou, D. Chakraborty, and K. Aberer. Sustainable energy consumption monitoring in residential settings. In *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on*, pages 1–6, Apr. 2013. doi: 10.1109/INFCOMW.2013.6562866.
- [60] R. Lukaszewski, K. Liszewski, and W. Winiecki. Methods of electrical appliances identification in systems monitoring electrical energy consumption. In *Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2013 IEEE 7th International Conference on*, volume 01, pages 10–14, Sep. 2013. doi: 10.1109/IDAACS.2013.6662630.
- [61] A. Zoha, A. Gluhak, M. Nati, and M. A. Imran. Low-power appliance monitoring using factorial hidden markov models. In *Intelligent Sensors, Sensor Networks and Information Processing, 2013 IEEE Eighth International Conference on*, pages 527–532, Apr. 2013. doi: 10.1109/ISSNIP.2013.6529845.
- [62] R. Jia, Y. Gao, and C. J. Spanos. A fully unsupervised non-intrusive load monitoring framework. In *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 872–878, Nov. 2015. doi: 10.1109/SmartGridComm.2015.7436411.
- [63] A. Ridi, C. Gisler, and J. Hennebert. Appliance and state recognition using hidden markov models. In *Data Science and Advanced Analytics (DSAA), 2014 International Conference on*, pages 270–276, Oct. 2014. doi: 10.1109/DSAA.2014.7058084.

- [64] J. Liang, S. K. K. Ng, G. Kendall, and J. W. M. Cheng. Load signature study part i: Basic concept, structure, and methodology. *IEEE Transactions on Power Delivery*, 25(2):551–560, Apr. 2010. ISSN 0885-8977.
- [65] J. Liang, S. K. K. Ng, G. Kendall, and J. W. M. Cheng. Load signature study part ii: Disaggregation framework, simulation, and applications. *IEEE Transactions on Power Delivery*, 25(2):561–569, Apr. 2010. ISSN 0885-8977. doi: 10.1109/TPWRD.2009.2033800.
- [66] A. Cole and A. Albicki. Nonintrusive identification of electrical loads in a three-phase environment based on harmonic content. In *Instrumentation and Measurement Technology Conference, 2000. IMTC 2000. Proceedings of the 17th IEEE*, volume 1, pages 24–29 vol.1, May 2000. doi: 10.1109/IMTC.2000.846806.
- [67] Kosuke Suzuki, Shinkichi Inagaki, Tatsuya Suzuki, Hisahide Nakamura, and Koichi Ito. Nonintrusive appliance load monitoring based on integer programming. In *SICE Annual Conference, 2008*, pages 2742–2747, Aug. 2008. doi: 10.1109/SICE.2008.4655131.
- [68] J. Li, S. West, and G. Platt. Power decomposition based on svm regression. In *Modelling, Identification Control (ICMIC), 2012 Proceedings of International Conference on*, pages 1195–1199, Jun. 2012.
- [69] S. R. Shaw, S. B. Leeb, L. K. Norford, and R. W. Cox. Nonintrusive load monitoring and diagnostics in power systems. *IEEE Transactions on Instrumentation and Measurement*, 57(7):1445–1454, Jul. 2008. ISSN 0018-9456. doi: 10.1109/TIM.2008.917179.
- [70] R. Cox, S. B. Leeb, S. R. Shaw, and L. K. Norford. Transient event detection for non-intrusive load monitoring and demand side management using voltage distortion. In *Twenty-First Annual IEEE Applied Power Electronics Conference and Exposition, 2006. APEC '06.*, pages 1751–1757, Mar. 2006. doi: 10.1109/APEC.2006.1620777.
- [71] H. T. Yang, H. H. Chang, and C. L. Lin. Design a neural network for features selection in non-intrusive monitoring of industrial electrical loads. In *2007 11th International Conference on Computer Supported Cooperative Work in Design*, pages 1022–1027, Apr. 2007. doi: 10.1109/CSCWD.2007.4281579.

- [72] Y. H. Lin and M. S. Tsai. A novel feature extraction method for the development of nonintrusive load monitoring system based on bp-ann. In *2010 International Symposium on Computer, Communication, Control and Automation (3CA)*, volume 2, pages 215–218, May 2010. doi: 10.1109/3CA.2010.5533571.
- [73] H. H. Chang, P. C. Chien, L. S. Lin, and N. Chen. Feature extraction of non-intrusive load-monitoring system using genetic algorithm in smart meters. In *e-Business Engineering (ICEBE), 2011 IEEE 8th International Conference on*, pages 299–304, Oct. 2011. doi: 10.1109/ICEBE.2011.48.
- [74] Hsueh-Hsien Chang, Hong-Tzer Yang, and Ching-Lung Lin. *Load Identification in Neural Networks for a Non-intrusive Monitoring of Industrial Electrical Loads*, pages 664–674. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. ISBN 978-3-540-92719-8. doi: 10.1007/978-3-540-92719-8_60. URL http://dx.doi.org/10.1007/978-3-540-92719-8_60.
- [75] K. D. Lee, S. B. Leeb, L. K. Norford, P. R. Armstrong, J. Holloway, and S. R. Shaw. Estimation of variable-speed-drive power consumption from harmonic content. *IEEE Transactions on Energy Conversion*, 20(3):566–574, Sep. 2005. ISSN 0885-8969. doi: 10.1109/TEC.2005.852963.
- [76] W. Wichakool, A. T. Avestruz, R. W. Cox, and S. B. Leeb. Modeling and estimating current harmonics of variable electronic loads. *IEEE Transactions on Power Electronics*, 24(12):2803–2811, Dec. 2009. ISSN 0885-8993. doi: 10.1109/TPEL.2009.2029231.
- [77] Hsueh-Hsien Chang. Non-intrusive demand monitoring and load identification for energy management systems based on transient feature analyses. *Energies*, 5(11): 4569, 2012. ISSN 1996-1073. doi: 10.3390/en5114569. URL <http://www.mdpi.com/1996-1073/5/11/4569>.
- [78] H. Y. Lam F. H. Y. Chan M. Lucente W. K. Lee, G. S. K. Fung. Exploration on load signatures. In *Proceeding of International Conference on Electrical Engineering (ICEE)*, pages 1–5, Sapporo, Japan, Jul. 2004.
- [79] <http://athemium.org/>.
- [80] <http://www.netvox.com.tw/Z-800.asp>.

- [81] D.L. Olson and D. Delen. *Advanced Data Mining Techniques*. Springer, 2008.
- [82] Michail G. Lagoudakis. The 0-1 knapsack problem – an introductory survey. Technical report, 1996.
- [83] James C. Bean. Multiple choice Knapsack functions. Technical report, Department of Industrial and Operations Engineering, The University of Michigan, Ann, Arbor, MI 48109-2117, January 1988.
- [84] Silvano Martello and Paolo Toth. Solution of the zero-one multiple knapsack problem. *European Journal of Operational Research*, 4(4):276 – 283, 1980. ISSN 0377-2217. doi: [http://dx.doi.org/10.1016/0377-2217\(80\)90112-5](http://dx.doi.org/10.1016/0377-2217(80)90112-5). URL <http://www.sciencedirect.com/science/article/pii/0377221780901125>. Combinational Optimization.
- [85] M.E. Dyer, N. Kayal, and J. Walker. A branch and bound algorithm for solving the multiple-choice knapsack problem. *Journal of Computational and Applied Mathematics*, 11(2):231 – 249, 1984. ISSN 0377-0427. doi: [http://dx.doi.org/10.1016/0377-0427\(84\)90023-2](http://dx.doi.org/10.1016/0377-0427(84)90023-2). URL <http://www.sciencedirect.com/science/article/pii/0377042784900232>.
- [86] Ulungu-Ekunda Lukata and Jacques Teghem. *Solving Multi-Objective Knapsack Problem by a Branch-and-Bound Procedure*, pages 269–278. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997. ISBN 978-3-642-60667-0. doi: 10.1007/978-3-642-60667-0_26.
- [87] A. S. Anagun and T. Sarac. *Optimization of Performance of Genetic Algorithm for 0-1 Knapsack Problems Using Taguchi Method*, pages 678–687. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-34076-8. doi: 10.1007/11751595_72. URL http://dx.doi.org/10.1007/11751595_72.
- [88] Alok Singh and Anurag Singh Baghel. A new grouping genetic algorithm for the quadratic multiple knapsack problem. In *Proceedings of the 7th European Conference on Evolutionary Computation in Combinatorial Optimization*, EvoCOP’07, pages 210–218, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 978-3-540-71614-3. URL <http://dl.acm.org/citation.cfm?id=1761927.1761946>.
- [89] A. S. Fukunaga. A new grouping genetic algorithm for the multiple knapsack problem. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress*

- on *Computational Intelligence*), pages 2225–2232, Jun. 2008. doi: 10.1109/CEC.2008.4631094.
- [90] W. Shen, B. Xu, and J. p. Huang. An improved genetic algorithm for 0-1 knapsack problems. In *2011 Second International Conference on Networking and Distributed Computing*, pages 32–35, Sept 2011. doi: 10.1109/ICNDC.2011.14.
- [91] P. Toth. Dynamic programming algorithms for the zero-one knapsack problem. *Computing*, 25(1):29–45. ISSN 1436-5057. doi: 10.1007/BF02243880. URL <http://dx.doi.org/10.1007/BF02243880>.
- [92] H. Shojaei, A. Ghamarian, T. Basten, M. Geilen, S. Stuijk, and R. Hoes. A parameterized compositional multi-dimensional multiple-choice knapsack heuristic for CMP run-time management. In *46th ACM/IEEE Design Automation Conference (DAC '09)*, pages 917–922, Jul. 2009. doi: 10.1145/1629911.1630147.
- [93] M. Geilen and T. Basten. A calculator for pareto points. In *2007 Design, Automation Test in Europe Conference Exhibition*, pages 1–6, Apr. 2007. doi: 10.1109/DATE.2007.364605.
- [94] Marc Geilen, Twan Basten, Bart D. Theelen, and Ralph Otten. An Algebra of Pareto Points. In *Int. Conf. on Application of Concurrency to System Design*, pages 88–97, 2005. doi: 10.1109/ACSD.2005.2.
- [95] Marc Geilen, Twan Basten, Bart D. Theelen, and Ralph Otten. An algebra of pareto points. *Fundamenta Informaticae*, 78:35–74, Sep. 2007.
- [96] A. Sbihi M. Hifi, M. Michrafy. Heuristic algorithms for the multiple-choice multidimensional knapsack problem. *The Journal of the Operational Research Society*, 55(12):1323–1332, 2004. ISSN 01605682, 14769360.
- [97] Michael A. Yukish. *Algorithms to Identify Pareto Points in Multi-dimensional Data Sets*. PhD thesis, 2004. AAI3148694.
- [98] Silvano Martello, David Pisinger, and Paolo Toth. Dynamic programming and strong bounds for the 0-1 knapsack problem. *Management Science*, 45(3):414–424, 1999. doi: 10.1287/mnsc.45.3.414. URL <http://dx.doi.org/10.1287/mnsc.45.3.414>.