Smart meter deployment optimisation and its analysis for appliance load monitoring

Ahmed Shaharyar Khwaja, Muhammad Naeem, Alagan Anpalagan, Tas A. Venetsanopoulos, Bala Venkatesh

Department of ELCE, Ryerson University, Toronto, Canada ON M5B 2K3
E-mail: alagan@ee.ryerson.ca

Published in *The Journal of Engineering*; Received on 11th December 2014; Accepted on 9th January 2015

Abstract: In this study, the authors study the problem of smart meter deployment optimisation for appliance load monitoring, that is, to monitor a number of devices without any ambiguity using the minimum number of low-cost smart meters. The importance of this problem is due to the fact that the number of meters should be reduced to decrease the deployment cost, improve reliability and decrease congestion. In this way, in future, smart meters can provide additional information about the type and number of distinct devices connected, besides their normal functionalities concerned with providing overall energy measurements and their communication. The authors present two exact smart meter deployment optimisation algorithms, one based on exhaustive search and the other based on efficient implementation of the exhaustive search. They formulate the problem mathematically and present computational complexity analysis of their algorithms. Simulation scenarios show that for a typical number of home appliances, the efficient search method is significantly faster compared to the exhaustive search and can provide the same optimal solution. The authors also show the dependency of their method on the distribution of the load pattern that can potentially be in a typical household.

Nomenclature

N	number of devices
L	number of meters
M	2^{N-1}
$\boldsymbol{A}$	An $M \times N$ matrix containing all possible M combinations of devices that are active (1) or inactive (0). Each row of $\boldsymbol{A}$ shows the devices that are active at the same time and connected to the same meter
$\boldsymbol{p}$	An $N \times 1$ vector containing the average power of each device
$\Delta\boldsymbol{p}$	An $N \times 1$ vector containing power deviation for each device, that is, the power of the device can vary from $p-\Delta p$ to $p+\Delta p$
$\boldsymbol{z}=\boldsymbol{Ap}$	An $N \times 1$ vector containing average power for all possible device combinations
$\Delta\boldsymbol{z}=\boldsymbol{A}\Delta\boldsymbol{p}$	An $N \times 1$ vector containing power deviation for all possible device combinations
$\tilde{\boldsymbol{A}}$	An $L \times N$ matrix whose rows are the number of meters and whose columns are the number of devices
z_k	kth element of $\boldsymbol{z}$
Δz_k	kth element of $\Delta\boldsymbol{z}$
$\tilde{\boldsymbol{a}}_i$	ith row of $\tilde{\boldsymbol{A}}$
$\tilde{\boldsymbol{a}}^j$	jth column of $\tilde{\boldsymbol{A}}$
$\tilde{a}_i^j$	jth element of $\tilde{\boldsymbol{a}}_i$
$\boldsymbol{A}^{\mathrm{T}}$	transpose of $\boldsymbol{A}$
$\boldsymbol{I}$	identity matrix

1 Introduction

Increasing energy demand and dwindling natural resources have put the energy demand on improving its efficiency. This has led to active research on finding solutions in which the users and generators are aware of where and how the power is being used [1]. The research has provided ways to improve the current power grid, which carries power from central generators to end users. The smart grid is one such improvement, allowing for two-way flow of electricity and information, compared with the current grid that is based on one-way flow of electricity [2]. Smart grid integrates communication technology and monitoring devices in the traditional grid. This allows for the provision of real-time energy consumption to the utility office as well as the users [3]. Smart meter is one such monitoring device in a smart grid system that can allow active participation of the consumer to manage power delivery and reduce its cost [4].

A smart meter is a digital energy meter that measures energy consumption and enables two-way and real-time communication between the consumers and the generator [5]. The smart meters currently available in the market can only provide the cumulative energy consumed by a whole house [6]. However, to make time-of-day usage charges and demand response more effective, individual appliance load monitoring is needed. The smart meters should have cognitive capabilities that allow them to determine which devices contribute to which consumption [6]. This capability can help the consumer participate fruitfully in the demand response programme. In this programme, a user may be contacted by the utility office to reduce his/her electricity consumption to save money. If the user knows how much each device is contributing to the electricity consumption, he/she may be able to correctly select and switch off the devices that would lead to a reduction in the electricity consumption.

2 Related work

Two research approaches for appliance load monitoring have emerged: the first approach is to use total signal information collected from meters, and then identify individual devices from the total signals based on signal processing and pattern recognition approaches. This approach requires collecting signatures of the devices [7] and extensive prior learning to recognise a load signature and to maintain a database [8]. This is known as non-intrusive load monitoring. Research on load monitoring using the power meter readings from a whole household was carried out in [9–20]. These approaches either work with high sampling rates, which require expensive hardware and cannot distinguish loads having similar transient behaviour or have poor performance in the presence of certain kinds of loads [20].

The second approach (called naive approach later in this paper) is to connect low-cost smart meters to each appliance in the household. These low-cost meters are energy monitoring devices with communication ability. However, such a strategy would lead to deployment of a large number of meters, leading to increasing costs,

congestion and decreased reliability. Consequently, research on deploying a number of low-cost smart meters less than the total number of devices was presented in [21, 22]. In these references, it is shown that taking into account load patterns for different devices, we can deploy smart meters using an approximate approach. This approach leads to a number of meters that can be no more than twice the number of meters obtained using the exact approach. The exact approach consists of checking each device individually for similarity between measurements with any other device or a set of devices. Devices with similar measurements are connected to separate meters, which enables unambiguous monitoring of devices. This method neither requires any training based on load patterns nor requires deployment of high-cost high-frequency smart meters. However, the computational time becomes very high with increasing number of devices and it is important to reduce the computational time for developing realistic and practically realisable solutions.

In this paper, we present a smart meter deployment optimisation scheme. The novel contributions of our work are:

- We present the smart meter deployment problem as a constrained optimisation problem. We show that this optimisation problem can be seen as minimising the number of rows of a matrix such that a number of constraints are satisfied. This matrix shows the connections between different devices and meters.
- We first present an exact search method, and show that it can be divided into two problems. We then present an efficient approach to solve the first problem. The approach does not make use of any approximation.
- We compare the number of computations required by exact search and efficient approaches, and show that the efficient approach requires less computations.
- We show by means of simulations that our approach takes less time and has the same performance compared to the exact approach. We further show the dependency of our method on the type of distribution of the load pattern.

We would like to emphasise that the proposed algorithm can be used for deploying low-cost smart meters in such a way that their normal functions related to energy measurements and communications are further supplemented by providing information about the types and number of devices that are active at a certain time. This is applicable in the future deployment of smart meters with enhanced capabilities.

3 Smart meter deployment optimisation problem formulation

We use $\boldsymbol{X}$, $\boldsymbol{x}$ and x to represent a matrix, a vector and an element of a vector, respectively. When $x_i \geq 0$ for all components i of a vector $\boldsymbol{x}$, we use $\boldsymbol{x} \geq 0$. The function diag($\tilde{\boldsymbol{a}}$) converts the vector $\tilde{\boldsymbol{a}}$ to a diagonal matrix, that is, $\tilde{\boldsymbol{a}}\boldsymbol{I}$.

3.1 Explanation of main variables

Let $\boldsymbol{A}$ be a matrix that shows all possible combinations of devices that are connected together to the same meter. A '1' indicates that a device is switched active, whereas a '0' indicates that it is switched inactive. The size of $\boldsymbol{A}$ in $M \times N$, where $M = 2^N - 1$. The size of $\boldsymbol{A}$ is $M \times N$. The mean power corresponding to N devices is shown by a vector $\boldsymbol{p}$, whereas $\Delta\boldsymbol{p}$ shows the power deviation corresponding to N devices. The maximum and minimum powers can vary from $\boldsymbol{p} + \Delta\boldsymbol{p}$ to $\boldsymbol{p} - \Delta\boldsymbol{p}$. The vectors $\boldsymbol{p}$ and $\Delta\boldsymbol{p}$ have dimensions $N \times 1$. The mean power corresponding to all possible combinations of devices connected together is shown by $\boldsymbol{z}$, whereas $\Delta\boldsymbol{z} = \boldsymbol{A}\Delta\boldsymbol{p}$ shows the power deviation corresponding to all possible combinations of devices connected together. These vectors have dimensions $M \times 1$.

3.2 Aim of the optimisation problem

We intend to find the minimum number of meters that can be used to monitor N devices. The resulting solution to this problem can be expressed by matrix $\tilde{\boldsymbol{A}}$. It shows the combination of devices that are connected to each of the meters. Its size is $L \times N$, where L is the number of meters. Each row of the solution matrix $\tilde{\boldsymbol{A}}$ shows the devices connected to a single meter.

3.3 Example

The meters should be connected to devices in such a way that there is no ambiguity of measurements, that is, no combination of devices connected to the same meter should generate similar measurements. As an example, consider two devices having mean powers 3 and 10 with deviations ±2 and ±3, that is

$$\boldsymbol{p} = \begin{bmatrix} 3 \\ 10 \end{bmatrix}_{2\times 1} \text{ and } \Delta\boldsymbol{p} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}_{2\times 1} \tag{1}$$

The matrix showing all possible combinations of devices that can be connected to an individual meter is given as

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}_{3\times 2} \tag{2}$$

Now the mean power and power deviation are given as

$$\boldsymbol{z} = \begin{bmatrix} 3 \\ 10 \\ 13 \end{bmatrix}_{3\times 1} \text{ and } \Delta\boldsymbol{z} = \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix}_{3\times 1} \tag{3}$$

It is evident that the measured powers can vary from $\boldsymbol{z} - \Delta\boldsymbol{z}$ to $\boldsymbol{z} + \Delta\boldsymbol{z}$. Now when the second device is active or both devices are active, we cannot decide with a single meter which combination is active. This is so that the power values can overlap, for example, if the measured power is 10 units, this power corresponds to the second device and both devices are active at the same time. Therefore, we need to use a minimum of two meters in this case to measure power without any ambiguity. In this case, the selected monitoring matrix $\tilde{\boldsymbol{A}}$ is

$$\tilde{\boldsymbol{A}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}_{2\times 2} \tag{4}$$

3.4 Optimisation problem

As a general rule, if the devices connected to the ith meter are given by $\tilde{\boldsymbol{a}}_i$, then to ensure that there is no overlap with the measurement corresponding to the kth combination, the following test is carried out [21]

$$|\tilde{\boldsymbol{a}}_i\boldsymbol{p} - z_k| > (\tilde{\boldsymbol{a}}_i\Delta\boldsymbol{p} + \Delta z_k) \tag{5}$$

It can be seen that by minimising the number of rows of $\tilde{\boldsymbol{A}}$, we can minimise the number of meters, provided there is no overlapping of measurements in all the combinations of the devices that are connected to the meters. The resulting optimisation problem is the minimisation of the number of rows L of the matrix $\tilde{\boldsymbol{A}}$. It can be

formulated formally as

$$
\begin{aligned}
&\min \quad L \\
&\text{subject to:} \\
&\quad C1{:}|\tilde{\boldsymbol{a}}_i \boldsymbol{p} - z_k| > (\tilde{\boldsymbol{a}}_i \Delta \boldsymbol{p} + \Delta z_k), \\
&\qquad \forall k = 1,\ \ldots,\ 2^N,\ i = 1,\ \ldots,\ L \\
&\quad C2{:}\tilde{a}_i^j \in \{0,\ 1\}, \\
&\qquad \forall i = 1,\ \ldots,\ L,\ j = 1,\ \ldots,\ N \\
&\quad C3{:}1 \le L \le N \\
&\quad C4{:} \sum_{j=1}^{N} \tilde{a}_i^j \ge 1, \quad \forall i = 1,\ \ldots,\ L \\
&\quad C5{:} \sum_{i=1}^{L} \tilde{a}_i^j = 1, \quad \forall j = 1,\ \ldots,\ N
\end{aligned}
\tag{6}
$$

where the five constraints $C1$–$C5$ are described as follows:

- $C1$ ensures that there is no ambiguity for measurements corresponding to $\tilde{\boldsymbol{a}}_i$, where $\tilde{\boldsymbol{a}}_i$ corresponds to the ith row of $\tilde{\boldsymbol{A}}$.
- $C2$ ensures that the entries of $\tilde{\boldsymbol{A}}$ are either 0 or 1.
- $C3$ ensures that there is at least one meter and the maximum number of meters is not more than the total number of devices.
- $C4$ ensures that the ith meter is connected to at least one device.
- $C5$ ensures that the jth device is connected to only one meter and all devices are monitored by meters.

The above problem is a binary integer optimisation problem as all the variables are binary. The problem is non-linear because of the presence of $C1$. The presence of the non-linear constraint makes the problem complicated to solve as traditional binary integer programming methods cannot be applied. In the following section, we present a two-step solution to this problem.

Algorithm 1

Input: $\boldsymbol{A}, M, \boldsymbol{p}, \Delta\boldsymbol{p}$
$i \leftarrow 1$
while $i \le M$ **do**
  $\bar{\boldsymbol{a}} \leftarrow i^{th}$ *row of* $\boldsymbol{A}$
  $\bar{\boldsymbol{A}} \leftarrow$ *matrix selected from* $\boldsymbol{A}$*, containing only those devices that are active in* $\bar{\boldsymbol{a}}$.
  $\bar{\boldsymbol{z}} \leftarrow$ *mean power of combination of devices active in* $\bar{\boldsymbol{A}}$
  $\Delta\bar{\boldsymbol{z}} \leftarrow$ *power deviation of combination of devices active in* $\bar{\boldsymbol{A}}$
  $\bar{n}_a \leftarrow$ *total number of rows in* $\bar{\boldsymbol{A}}$
  $j \leftarrow 1$
  $sum \leftarrow 0$
  while $j \le \bar{n}_a$ **do**
    $\hat{\boldsymbol{a}} \leftarrow j^{th}$ *row of* $\bar{\boldsymbol{A}}$
    $\hat{\boldsymbol{z}} \leftarrow$ *mean power of combinations in* $\bar{\boldsymbol{A}}$*, excluding* $\hat{\boldsymbol{a}}$
    $\Delta\hat{\boldsymbol{z}} \leftarrow$ *power deviation for combinations in* $\bar{\boldsymbol{A}}$*, excluding* $\hat{\boldsymbol{a}}$
    if $|\hat{\boldsymbol{a}}\boldsymbol{p} - \hat{\boldsymbol{z}}| > (\hat{\boldsymbol{a}}\Delta\boldsymbol{p} + \Delta\hat{\boldsymbol{z}})$ **then**
      $sum \leftarrow sum + 1$
    end if
    $j \leftarrow j + 1$
  end while
  if $sum = \bar{n}_a$ **then**
    $\tilde{a}_i \leftarrow 1$
  else
    $\tilde{a}_i \leftarrow 0$
  end if
  $i \leftarrow i + 1$
end while
$\tilde{\boldsymbol{A}}_t \leftarrow (\boldsymbol{A}^T \mathrm{diag}(\tilde{\boldsymbol{a}}))^T$
Output: $\tilde{\boldsymbol{A}}_t$

Fig. 1 *Brute force solution for smart meter deployment problem*

doi: 10.1049/joe.2014.0345

4 Solution of the optimisation problem

4.1 Brute force solution

A brute force solution to the problem formulated in (6) can be obtained by checking the measured power corresponding to the selected combination in $\boldsymbol{A}$ against power corresponding to all the remaining combinations in $\boldsymbol{A}$. Any combination that gives ambiguity in measurement is removed. The combination that results in the minimum number of meters is retained. The implementation of the solution can be divided into two steps:

Step 1: Identification of combinations without conflicts: The implementation of the first step is shown in Algorithm 1 (see Fig. 1). This algorithm consists of M iterations. At each iteration, the following main steps are carried out:

- One row of $\boldsymbol{A}$ is selected, represented by vector $\bar{\boldsymbol{a}}$.
- To avoid unnecessary computations, only those combinations that have devices active in the current combination are subsequently selected.
- The mean power and power deviations for the combinations in $\bar{\boldsymbol{A}}$ are calculated and stored in vectors $\bar{\boldsymbol{z}}$ and $\Delta\bar{\boldsymbol{z}}$, respectively.
- The measured power corresponding to $\bar{\boldsymbol{a}}$ is compared against measured powers given in $\bar{\boldsymbol{A}}$. This comparison is based on (5). This is carried out in lines 11–19 of (Fig. 1).
- If the condition given by (5) is not violated, $\bar{\boldsymbol{a}}$ is selected, otherwise it is discarded. This selection is given by vector $\tilde{\boldsymbol{a}}$ that has a size of $M \times 1$, and the ith element $\tilde{a}_i$ denotes the combination number. This is shown in lines 20–24. Those combinations that do not have any measurement ambiguities are stored in $\tilde{\boldsymbol{A}}_t$. This is carried out in line 27. The result $\tilde{\boldsymbol{A}}_t$ is used as input to the next algorithm for selection of combinations such that the number of meters is minimised.

Step 2: Minimising the number of meters: The second step is given in Algorithm 2 (see Fig. 2). In this algorithm, a combination of the rows of $\tilde{\boldsymbol{A}}_t$ is selected such that the number of rows L is minimum and constraints $C3$–$C5$ are satisfied. The algorithm consists of the following steps:

Algorithm 2

1: **Input:** $\tilde{\boldsymbol{A}}_t, N$
2: $\tilde{n}_a \leftarrow$ *number of rows of* $\tilde{\boldsymbol{A}}_t$
3: $i \leftarrow 1$
4: **while** $i \leq \tilde{n}_a$ **do**
5: $\quad \tilde{N}_i \leftarrow \frac{\tilde{n}_a!}{i!(\tilde{n}_a - i)!}$
6: $\quad \nu \leftarrow$ *all possible combinations for values* $1, 2, ..., \tilde{n}_a$, ν *has* $\tilde{N}_i$ *rows and* i *columns*
7: $\quad j \leftarrow 1$
8: $\quad$ **while** $j \leq \tilde{N}_i$ **do**
9: $\quad\quad \boldsymbol{C} \leftarrow$ *select rows of* $\tilde{\boldsymbol{A}}_t$ *corresponding to the* j^{th} *combination in* ν
10: $\quad\quad sum \leftarrow 0$
11: $\quad\quad k \leftarrow 1$
12: $\quad\quad$ **while** $k \leq N$ **do**
13: $\quad\quad\quad$ **if** $\sum_{l=1}^{\tilde{N}_i} c_l^k = 1$ **then**
14: $\quad\quad\quad\quad sum \leftarrow sum + 1$
15: $\quad\quad\quad$ **end if**
16: $\quad\quad\quad$ **if** $sum = N$ **then**
17: $\quad\quad\quad\quad \tilde{\boldsymbol{A}} \leftarrow \boldsymbol{C}$
18: $\quad\quad\quad\quad L \leftarrow$ *number of rows of* $\tilde{\boldsymbol{A}}$
19: $\quad\quad\quad\quad$ return
20: $\quad\quad\quad$ **end if**
21: $\quad\quad\quad k \leftarrow k + 1$
22: $\quad\quad$ **end while**
23: $\quad\quad j \leftarrow j + 1$
24: $\quad$ **end while**
25: $\quad i \leftarrow i + 1$
26: **end while**
27: **Output:** $\tilde{\boldsymbol{A}}$

Fig. 2 *Combination selection*

Algorithm 3

1: **Input:** $\boldsymbol{A}, N, p, \Delta p$
2: $i \leftarrow 1$
3: **while** $i \leq N$ **do**
4: $\bar{\boldsymbol{A}} \leftarrow$ *combinations in* $\boldsymbol{A}$ *that have* i *non-zero bits*
5: $\bar{n}_a \leftarrow$ *number of rows in* $\bar{\boldsymbol{A}}$
6: $j \leftarrow 1$
7: **while** $j \leq \bar{n}_a$ **do**
8: $\bar{\boldsymbol{a}} \leftarrow j^{th}$ *row of* $\bar{\boldsymbol{A}}$
9: $\hat{\boldsymbol{A}} \leftarrow$ *all the combinations having non-zero bits in all or the same position as in* $\bar{\boldsymbol{a}}$
10: $\hat{n}_a \leftarrow$ *number of rows in* $\hat{\boldsymbol{A}}$
11: **if** $\hat{n}_a > 1$ **then**
12: $\boldsymbol{z} \leftarrow \hat{\boldsymbol{A}}\boldsymbol{p}$
13: $\Delta \boldsymbol{z} \leftarrow \hat{\boldsymbol{A}}\Delta \boldsymbol{p}$
14: $z_c \leftarrow \bar{\boldsymbol{a}}\boldsymbol{p}$
15: $\Delta z_c \leftarrow \bar{\boldsymbol{a}}\Delta \boldsymbol{p}$
16: $k \leftarrow 1$
17: **while** $k \leq \hat{n}_a$ **do**
18: **if** $|z_c - z_j| \leq (\Delta z_c + \Delta z_j)$ **then**
19: $\boldsymbol{A} \leftarrow$ *remove all rows that contain the conflicted combination given by* $\bar{\boldsymbol{a}}$
20: **end if**
21: $k \leftarrow k + 1$
22: **end while**
23: **end if**
24: $j \leftarrow j + 1$
25: **end while**
26: $\bar{\boldsymbol{A}} \leftarrow$ *those combinations in* $\boldsymbol{A}$ *that have* i *non-zero bits*
27: **if** $\bar{\boldsymbol{A}}$ *is not empty* **then**
28: $\bar{n}_a \leftarrow$ *number of rows in* $\bar{\boldsymbol{A}}$
29: $\boldsymbol{z} \leftarrow \bar{\boldsymbol{A}}\boldsymbol{p}$
30: $\Delta \boldsymbol{z} \leftarrow \bar{\boldsymbol{A}}\Delta \boldsymbol{p}$
31: $j \leftarrow 1$
32: **while** $j \leq \bar{n}_a$ **do**
33: $k \leftarrow j + 1$
34: **while** $k \leq \bar{n}_a$ **do**
35: **if** $|z_j - z_k| \leq (\Delta z_j + \Delta z_k)$ **then**
36: $\boldsymbol{a}' \leftarrow$ *combine* j^{th} *row in* $\bar{\boldsymbol{A}}$ *with* k^{th} *row in* $\bar{\boldsymbol{A}}$
37: $\boldsymbol{A} \leftarrow$*remove all rows that contain the conflicted combination given by* $\boldsymbol{a}'$
38: **end if**
39: $k \leftarrow k + 1$
40: **end while**
41: $j \leftarrow j + 1$
42: **end while**
43: **end if**
44: $i \leftarrow i + 1$
45: **end while**
46: $\tilde{\boldsymbol{A}}_t \leftarrow \boldsymbol{A}$
47: **Output:** $\tilde{\boldsymbol{A}}_t$

Fig. 3 *Efficient solution*

- Different number of rows from $\tilde{\boldsymbol{A}}_t$ are selected to make a combination. The number of rows is increased gradually, starting from 1, which means that we first check if it is possible to monitor all the devices using a single meter. If this is not possible, we check for two-row combinations of $\tilde{\boldsymbol{A}}_t$, that is, we check if all the devices can be monitored using two meters. The process of generating single and multiple row combinations in $\tilde{\boldsymbol{A}}_t$ is shown by lines 4–9. The variable i defines the number of rows that should be considered together. The total number of such constraints is given by $\tilde{N}_i$ and ν contains the row number of these combinations. The matrix $\boldsymbol{C}$ contains each of the combinations in $\tilde{\boldsymbol{A}}_t$ that should be checked for constraint satisfactions, for example, if $i = 1$, each row of $\tilde{\boldsymbol{A}}_t$ will be checked to see if constraints $C3$–$C5$ are satisfied. Similarly, $i = 2$ means that two rows at most will be checked for constraint satisfaction to see if any one of them satisfies the constraints.
- Lines 12–24 check if constraints $C3$–$C5$ are satisfied. Line 13 checks $C4$ and $C5$, while specifically line 16 checks $C3$.
- Once a constraint is satisfied by a combination, it is stored in $\tilde{\boldsymbol{A}}$ and the number of rows is stored in L. The programme is stopped once all the constraints are satisfied.

4.2 *Efficient solution*

The brute force algorithm is computationally expensive. For a large number of devices, the number of computations can be very large requiring significant computing resources. To deal with this issue, we propose an efficient algorithm. The details of this algorithm are shown in Algorithm 3 (see Fig. 3). The variable i refers to the number of active devices considered in one iteration. The matrix $\bar{\boldsymbol{A}}$ denotes the selected combinations that have i active devices, whereas $\bar{n}_a$ is the total number of combinations present in $\bar{\boldsymbol{A}}$. The description of the above-mentioned two steps is as follows:

Step 1: Verifying lower level combinations: This step is shown in lines 7–26 that carries out iterations ranging from 1 to the total number of combinations in $\bar{\boldsymbol{A}}$. Different steps involved in this algorithm are as follows:

- The jth row of $\boldsymbol{A}$ is stored in $\bar{\boldsymbol{a}}$.
- The matrix $\hat{\boldsymbol{A}}$ contains all the combinations that have at least one and up to the total number of active devices in $\bar{\boldsymbol{a}}$.
- The measurements corresponding to $\bar{\boldsymbol{a}}$ are stored in z_c and Δz_c and the measurements corresponding to combinations in $\hat{\boldsymbol{A}}$ are stored in vectors $\boldsymbol{z}$ and $\Delta \boldsymbol{z}$. The measurement corresponding to $\bar{\boldsymbol{a}}$ is compared against the measurements corresponding to each combination in $\hat{\boldsymbol{A}}$. This is carried out in lines 17–23.
- If the measurement corresponding to $\bar{\boldsymbol{a}}$ has a conflict with any of the combinations in $\hat{\boldsymbol{A}}$, the combination corresponding to $\bar{\boldsymbol{a}}$ is removed from $\boldsymbol{A}$, for example, if $\bar{\boldsymbol{a}} = [0 \quad 0 \quad 1 \quad 1]$, that is, the third and fourth devices are active, we compare it with combinations in $\hat{\boldsymbol{A}}$ that are

$$\hat{\boldsymbol{A}} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}_{2\times 4} \tag{7}$$

The comparison of $\bar{\boldsymbol{a}}$ with $\hat{\boldsymbol{A}}$ checks if there is a conflict at the lower level, that is, the devices that make up the combination $\bar{\boldsymbol{a}}$. If there is a conflict, it means that any combination of devices that contains first and second devices in active positions should be removed, for example, if $\bar{\boldsymbol{a}}$ gives a conflict, then [0 1 1 1] and [1 0 1 1] are not feasible combinations and are removed from $\boldsymbol{A}$. Once all the combinations in $\hat{\boldsymbol{A}}$ are verified, the second step of the algorithm is carried out.

Step 2: Verifying higher-level combinations: In this step, all the combinations that have i non-zero active devices are selected in $\bar{\boldsymbol{A}}$

doi: 10.1049/joe.2014.0345

and the combinations are checked to see if they have any conflicts in measurements. This is carried out in lines 28–44. Following are the main steps:

- The measurements corresponding to $\bar{A}$ are stored in vectors z and Δz.
- Lines 33–43 compare measurement corresponding to each measurement in $\bar{A}$ with the measurements corresponding to remaining combinations in $\bar{A}$.
- If there is a conflict, the combinations giving conflicting measurements are combined in vector a'.
- All the combinations having the conflicting measurements given by a' are removed from A. This ensures that any higher-level combination that is made up of devices including the conflicted combination is removed. In fact, this step removes any conflicting higher-level combination based on a', for example, consider a two-device combination $\bar{A}$ given as

$$\bar{A} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}_{6\times 4} \tag{8}$$

Each row in $\bar{A}$ is compared with all the remaining rows in $\bar{A}$. If there is a conflict, all the rows in A that have the active devices giving the conflict are removed from A, for example, if $\bar{a}_1 = [0 \quad 0 \quad 1 \quad 1]$ has a conflict with $\bar{a}_2 = [0 \quad 1 \quad 0 \quad 1]$, all the rows in A having first, second and third devices as active are removed from A. The remaining combinations are stored in a matrix $\tilde{A}_t$.

Step 3: Minimising the number of meters: The resulting matrix A_t from the previous step is used in Fig. 2 to select a combination that minimises the number of meters.

4.3 Computational complexity

The number of computations can be calculated by considering that if there are k number of active devices, the total number of combinations that can have 1 to k number of active devices is given as

$$M_k = \frac{N!}{k!(N-k)!} \tag{9}$$

We further consider the number of comparisons as a criterion for computational complexity.

1. *Brute force method:* This can be considered as picking up one of the $2^N - 1$ combinations from A, identified henceforth as current combination.

- If there are k active devices in a current combination, the total number of combinations to be compared for overlapping measurements with the current combination is $2^k - 1$. Subsequently, the total number of comparisons is $2^k - 2 + 2^k - 3 + \cdots + 1$. For M_k combinations that contain k active devices, the total number of checks is $(2^k - 2 + 2^k - 3 + \cdots + 1) \times M_k$.
- The total number of comparisons is carried out for number of active devices ranging from 1 to N, that is, k varies from 1 to N. For single active device combinations, that is, $k = 1$, these comparisons are $(N - 1 + N - 2 + \cdots + 1)$ as there are N single active device combinations.
- The total number of comparisons is

$$\sum_{k=2}^{N} \sum_{n=2}^{2^k-1} (2^k - n) \times M_k + \sum_{n=1}^{N-1} (N - n)$$

2. *Efficient solution:* This can be considered as selecting a current combination with increasing the number of active devices and carrying out comparisons in two steps as detailed previously:

- When only combinations with a single active device are selected, there is only the first kind of comparison. The number of comparisons is $N - 1$ for the first selected combination, $N - 2$ for the second selected combination and so on. The total number of such comparisons is $N - 1 + N - 2 + \cdots + 1$.
- Starting from two and more than two active devices, the total number of such comparisons can be calculated by considering two stages:
  - (i) If k is the number of active devices, the total number of combinations that contain 1 or all the k devices activated is given by $2^k - 1$. Subsequently, the comparisons in the first stage are given by $(2^k - 2) \times M_k$.
  - (ii) The comparisons in the second stage are given by $M_k - 1 + M_k - 2 + \cdots + 1$.
- When the total number of active devices is N, there is only the second kind of comparison. In this case, the number of comparisons is given by $(2^N - 2) \times N$, as the total number of such combinations is given by N.
- The total number of comparisons is

$$\sum_{k=2}^{N-1} (2^k - 2) \times M_k + \sum_{n=1}^{M_k-1} (M_k - n) + (2^N - 2) \times N$$

The computational complexity of the efficient solution is the worst-case scenario, as we do not consider the removal of any combinations that give conflicting measurements. This can lead to further reduction of computational complexity. It can be seen that the efficient solution leads to significant computational savings compared with the brute force method. The total number of computational savings is given as

$$\sum_{k=2}^{N} \sum_{n=3}^{2^k-1} (2^k - n) \times M_k - \sum_{n=1}^{M_k-1} (M_k - n)$$

5 Simulation results

In this section, we present simulation results for the smart meter deployment optimisation problem. We compare results obtained using the exhaustive search and efficient methods. We also compare the performance of the efficient search method with respect to different distributions.

5.1 Comparison of number of meters obtained using both methods

First, we compare both approaches with a selected load pattern. For the sake of clarity, the number of devices is chosen as $N = 4$, and the load pattern is illustrated in Fig. 4.

Fig. 1 based on exhaustive search consists of 15 iterations. The output of Fig. 1 is used by Fig. 2 to determine the solution. The output of the exhaustive search method is shown as follows

$$\tilde{A} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}_{2\times 4} \tag{10}$$

The result in Fig. 5 shows that in total two meters are required. Next, we show the results of the steps using the efficient search method. Fig. 3 consists of four iterations. At the end of the first

iteration, the result is given as

$$\tilde{\boldsymbol{A}}_t = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}_{8\times 4} \tag{11}$$

which means that seven combinations have been eliminated at this stage. This matrix is used in the next iterations. At the end of all the iterations, the result is

$$\tilde{\boldsymbol{A}}_t = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}_{7\times 4} \tag{12}$$

where the final result after application of Fig. 2 to $\tilde{\boldsymbol{A}}_t$ is the same as in (10).

Next, we simulate varying numbers of devices, ranging from 5 to 15, as this is a reasonable number for a typical home. We choose arbitrary load pattern and calculate the number of meters required to monitor the devices. We use both exhaustive search and efficient search methods and the results are shown in Table 1. It can be seen that both methods give similar numbers of meters ranging from 2 to 4, which means that the efficient search method is able to perform as well as exhaustive search. We also compare the savings in terms of price and number of meters obtained using our method. The estimated price of a smart meter is $200 [23], whereas based on the price of the energy meter [23], we estimate the price of a low-cost smart meter as $20. Note that this is a conservative estimate; the actual price for a low-cost smart meter can be <$20 [24]. The saving ratio is calculated as the ratio of the cost of the calculated number of low-cost smart meters to the cost of a single high-cost smart meter. The maximum saving ratio is 0.4, which means that we can at least save 60% deployment cost using our method. The number ratio with respect to naive approach is calculated as the ratio of the calculated number of smart meters to the total number of devices being monitored. The maximum number ratio is 0.4, which means that we can deploy at least 60% less meters compared with the naive approach, subsequently leading to a cost saving of 60%.

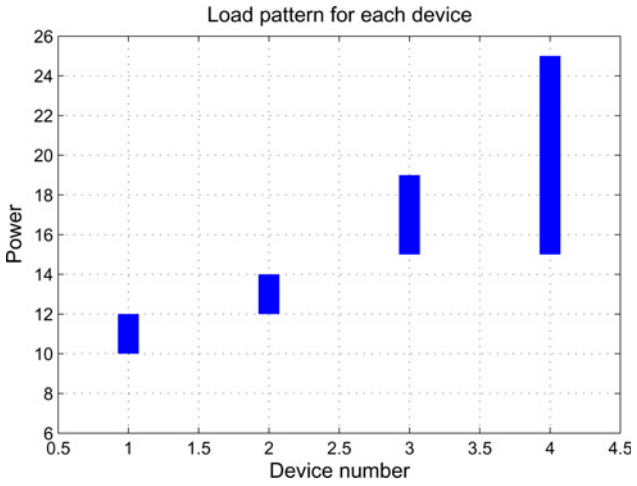

Fig. 4 *Load pattern for the example*

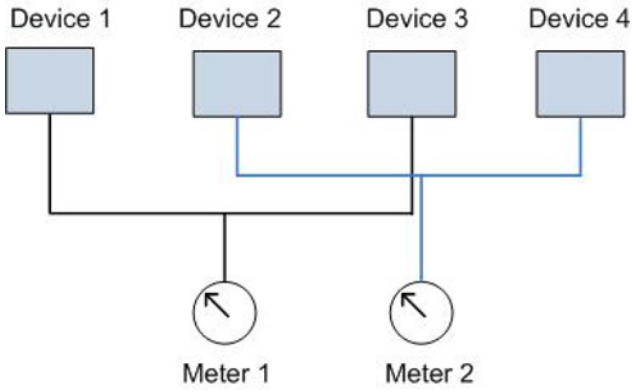

Fig. 5 *Connections corresponding to (10)*

5.2 Comparison of computation time compared with exhaustive search

Next, we compare the computational times of both approaches. The load patterns are chosen arbitrarily as in the preceding section. The time taken is averaged over a number of iterations and the result is shown in Fig. 6. It can be seen that the efficient method takes less time compared with the exhaustive search method. For a number of devices <13, the time taken by both approaches is approximately similar. However, for a large number of devices, the increase in computation time using the efficient approach is much less than that obtained using the exhaustive search approach. In fact, the efficient search is 100 times faster than the exhaustive search method, which shows that the efficient search approach can be used for practically optimising a larger number of devices.

5.3 Computational time with respect to different distributions

First, we select a load pattern having a normal distribution. Load patterns with such distributions are shown to be generated by actual measurements [25, 26]. The mean of the distribution is selected as 200, and the variance is chosen as $(200/3)^2$, $(100/3)^2$ and $(50/3)^2$. The results using the efficient search are shown in Fig. 7. It can be seen that the computation time increases with increasing variance. The reason is that with an increase of variance, the number of appliances with overlapping measurements decreases as the load patterns are chosen from a large range of values. As we have shown earlier, the exhaustive search removes combinations with overlapping measurements in each iteration. Hence, more devices are removed from the combination at each iteration, which reduces the number of comparisons in subsequent iterations, subsequently speeding up the process.

We also compare the results for uniformly and exponentially distributed load patterns. The latter distribution represents the case

Table 1 Comparison of existing references

Number of appliances	5	7	9	11	13	15
Total number of meters obtained using brute force method	2	2	3	3	3	4
Total number of meters obtained using efficient method	2	2	3	3	3	4
Saving ratio with respect to high-frequency smart meter	0.2	0.2	0.3	0.3	0.3	0.4
Number ratio with respect to naive approaches	0.4	0.29	0.33	0.27	0.23	0.27

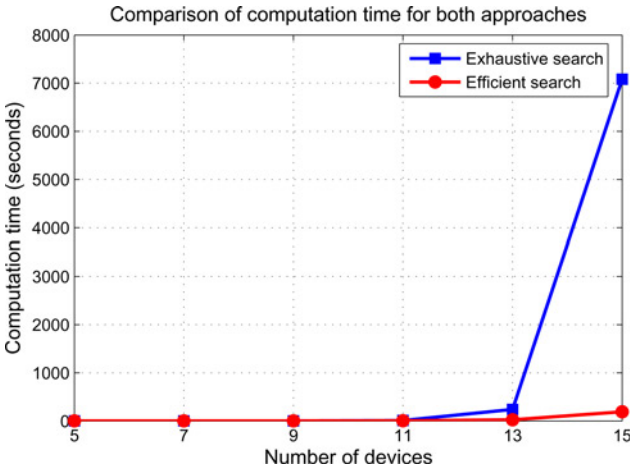

Fig. 6 *Comparison of computational time against number of devices using both methods*

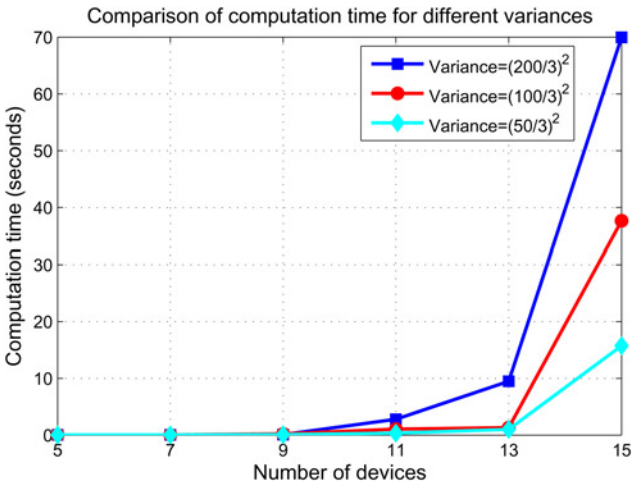

Fig. 8 *Computational time against number of devices for different variances in uniform distribution*

where the load patterns are very concentrated, whereas the former distribution represents the case of evenly distributed load patterns [22]. We chose these distributions as they represent opposite variations with respect to the normal distribution and can be used to show the effects of the degree of concentration of load patterns on the performance of the algorithms. Results similar to normal distribution are obtained when load patterns are simulated with uniform distribution, as shown in Fig. 8. However, in this case, the change in the computation time with respect to variance is less compared with the normal distribution. We further compare our results for load pattern distributed according to exponential distribution. The mean of the distribution is chosen as 200. Fig. 9 shows a comparison of the results with the three distributions. We can see that load patterns with normal distribution take the most time to obtain an optimisation solution, as in this case the values of different load patterns are distributed in a larger range. The times taken for normal and exponential load patterns are almost similar. For comparison, we show the computation time taken with the three distributions using exhaustive search in Fig. 10. As expected, the computation time compared with the efficient search is greater and similar trend can be observed as in Fig. 9 with respect to different distributions; however, in this case the computation time difference between different distributions is less. The reason may be that unlike the proposed efficient search that saves computations by iteratively removing combinations that can produce a conflict at subsequent iterations, the exhaustive search compares all the possible combinations to find the optimisation solution.

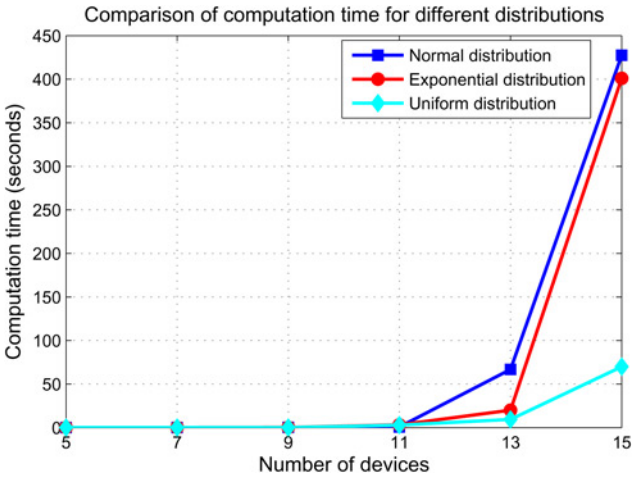

Fig. 9 *Computational time for efficient search against number of devices for different distributions*

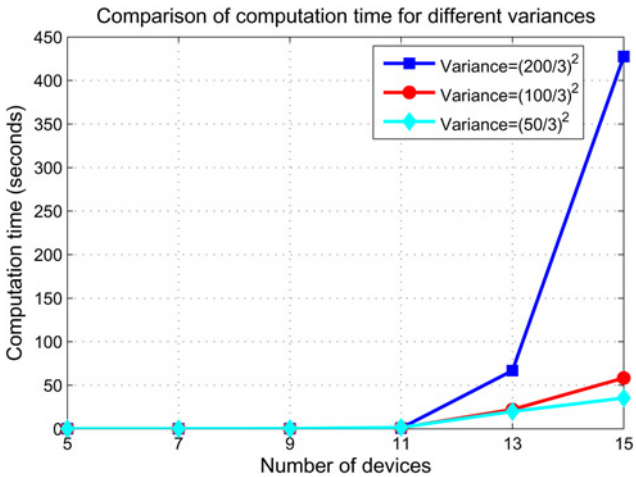

Fig. 7 *Computational time against number of devices for different variances in normal distribution*

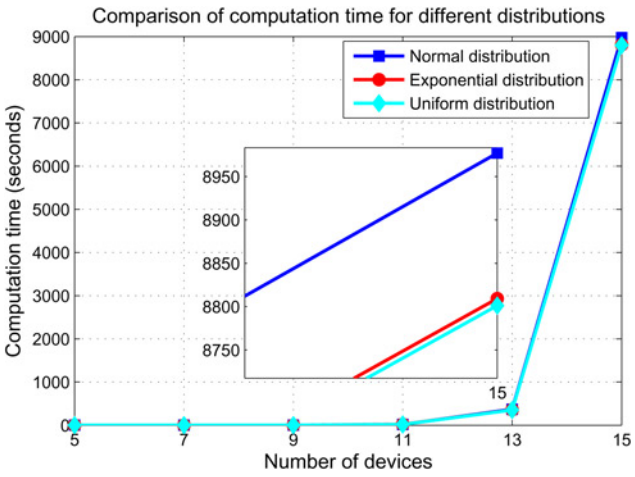

Fig. 10 *Computational time for exhaustive search against number of devices for different distributions*

6 Conclusions

In this paper, we examined the problem of smart meter deployment optimisation for appliance load monitoring. We formulated the problem mathematically and presented two algorithms to solve the problem: the first one is the exact exhaustive search method, whereas the second one is based on efficient implementation of the exact search method. We carried out a computational complexity analysis of the algorithms and showed that the efficient implementation requires less computation. We also carried out simulations that show that the performance of the efficient method is the same as the exact approach in terms of number of meters, and it takes less time to give a solution. We also showed the dependency of our method with respect to different load pattern distributions. We found out that load patterns with normal and exponential distributions lead to higher computation time.

7 References

[1] Heirman D.: 'What makes smart grid smart and who is in the 'game'', *IEEE Electromagn. Compat.*, 2012, **1**, (2), pp. 95–99

[2] Fang X., Misra S., Xue G., *ET AL.*: 'Smart grid – the new and improved power grid: a survey', *IEEE Commun. Surv. Tutor.*, 2012, **14**, (4), pp. 944–980

[3] Ye Y., Yi Q., Sharif H., *ET AL.*: 'A survey on smart grid communication infrastructures: motivations, requirements and challenges', *IEEE Commun. Surv. Tutor.*, 2013, **15**, (1), pp. 5–20

[4] Namboodiri V., Aravinthan V., Mohapatra S., *ET AL.*: 'Toward a secure wireless-based home area network for metering in smart grids', *IEEE Syst. J.*, 2014, **8**, (2), pp. 509–520

[5] Arif A., Al-Hussain M., Al-Mutairi N., *ET AL.*: 'Experimental study and design of smart energy meter for the smart grid'. Proc. IRSEC, 2013, pp. 515–520

[6] Makonin S., Popowich F., Gill B.: 'The cognitive power meter: looking beyond the smart meter'. Proc. IEEE CCECE, 2013

[7] Dong M., Meira P., Xu W., *ET AL.*: 'Non-intrusive signature extraction for major residential loads', *IEEE Trans. Smart Grid*, 2013, **4**, (3), pp. 1421–1430

[8] Ming D., Meira P., Wilsun X., *ET AL.*: 'An event window based load monitoring techniques for smart meters', *IEEE Trans. Smart Grid*, 2012, **3**, (2), pp. 787–796

[9] Sultanem F.: 'Using appliance signatures for monitoring residential loads at meter panel level', *IEEE Trans. Power Deliv.*, 1991, **6**, (4), pp. 1380–1385

[10] Hart G.: 'Nonintrusive appliance load monitoring', *Proc. IEEE*, 1992, **80**, (12), pp. 1870–1891

[11] Parson O., Ghosh S., Weal M., *ET AL.*: 'Non-intrusive load monitoring using prior models of general appliance types'. Proc. AAAI-12, 2012

[12] Kim H., Marwah M., Arlitt M., *ET AL.*: 'Unsupervised disaggregation of low frequency power measurements'. Proc. DSM, 2010

[13] Zeifman M., Roth K.: 'Nonintrusive appliance load monitoring: review and outlook', *IEEE Trans. Consum. Electron.*, 2011, **57**, (1), pp. 76–84

[14] Kolter J., Jaakkola T.: 'Approximate inference in additive factorial HMMS with application to energy disaggregation', *J. Mach. Learn. Res.*, 2012, **22**, pp. 1472–1482

[15] Cole A., Albicki A.: 'Nonintrusive identification of electrical loads in a three-phase environment based on harmonic content'. Proc. IEEE Instrumentation and Measurement Technology Conf., 2000, pp. 24–29

[16] Chang H., Chen K., Tsai Y., *ET AL.*: 'A new measurement method for power signatures of non-intrusive demand monitoring and load identification', *IEEE Trans. Ind. Appl.*, 2012, **48**, (2), pp. 764–771

[17] Laughman C., Kwangduk Lee L., Cox R., *ET AL.*: 'Power signature analysis', *IEEE Power Energy Mag.*, 2003, **1**, (2), pp. 56–63

[18] Leeb S., Shaw S., Kirtley J.Jr.: 'Transient event detection in spectral envelope estimates for nonintrusive load monitoring', *IEEE Trans. Power Deliv.*, 1995, **10**, (3), pp. 1200–1210

[19] Shaw S., Leeb S., Norford L., *ET AL.*: 'Nonintrusive load monitoring and diagnostics in power systems', *IEEE Trans. Instrum. Meas.*, 2008, **57**, (7), pp. 1445–1454

[20] Zoha A., Gluhak A., Imran M., *ET AL.*: 'Non-intrusive load monitoring approaches for disaggregated energy sensing: a survey', *Sensors*, 2012, **12**, pp. 16 838–16p 866

[21] Hao X., Wang Y., Wu C., *ET AL.*: 'Smart meter deployment optimization for efficient electrical appliance state monitoring'. Proc. IEEE SmartGridComm, 2012

[22] Wang Y., Hao X., Song L., *ET AL.*: 'Tracking states of massive electrical appliances by lightweight metering and sequence decoding'. Proc. SensorKDD, 2012

[23] Berges M., Goldman E., Matthews H., *ET AL.*: 'Enhancing electricity audits in residential buildings with nonintrusive load monitoring', *J. Ind. Ecol.*, 2010, **14**, (5), pp. 844–858

[24] Markham A., Danezisy G., Fu K., *ET AL.*: 'Designing privacy-preserving smart meters with low-cost microcontrollers', *Financ. Cryptogr. Data Sec. Lect. Notes Comput. Sci.*, 2012, **7397**, pp. 239–253

[25] Marwah M., Arbitt M., Lyon G., *ET AL.*: 'Unsupervised disaggregation of low frequency power measurements'. Technical Report, HP labs, 2010

[26] Sankar L., Rajagopalan S., Mohajer S., *ET AL.*: 'Smart meter privacy: a theoretical framework', *IEEE Trans. Smart Grid*, 2013, **4**, (2), pp. 837–846