An Appliance-driven Approach to Detection of Corrupted Load Curve Data

Guoming Tang[1,3], Kui Wu[1,4], Jian Pei[2], Jiuyang Tang[3], and Jingsheng Lei[4]
[1]University of Victoria, Victoria, BC, Canada
[2]Simon Fraser University, Burnaby, BC, Canada
[3]National University of Defense Technology, Changsha, Hunan, China
[4]Shanghai University of Electric Power, Shanghai, China
{guoming, wkui}@uvic.ca, jpei@sfu.ca, jiuyang_tang@nudt.edu.cn,
jshlei@shiep.edu.cn

ABSTRACT

Load curve data in power systems refers to users' electrical energy consumption data periodically collected with meters. It has become one of the most important assets for modern power systems. Many operational decisions are made based on the information discovered in the data. Load curve data, however, usually suffers from corruptions caused by various factors, such as data transmission errors or malfunctioning meters. To solve the problem, tremendous research efforts have been made on load curve data cleansing. Most existing approaches apply outlier detection methods from the supply side (*i.e.*, electricity service providers), which may only have aggregated load data. In this paper, we propose to seek aid from the demand side (*i.e.*, electricity service users). With the help of readily available knowledge on consumers' appliances, we present an appliance-driven approach to load curve data cleansing. This approach utilizes data generation rules and a Sequential Local Optimization Algorithm (SLOA) to solve the Corrupted Data Identification Problem (CDIP). We evaluate the performance of SLOA with real-world trace data and synthetic data. The results indicate that, comparing to existing load data cleansing methods, such as B-spline smoothing, our approach has an overall better performance and can effectively identify consecutive corrupted data. Experimental results also show that our method is robust in various tests.

Categories and Subject Descriptors

H.5.m [**Information Systems**]: Information interfaces and presentation—*Miscellaneous*

Keywords

Corrupted data identification; Data analysis; Optimization

1. INTRODUCTION

Electricity usage data, on the one hand, plays an important role in big data applications, and on the other hand, has been severely under explored. A recent news article appeared in Forbes [23] said,

"*But for the most part, utilities have yet to realize the potential of the flood of new data that has begun flowing to them from the power grid, …, And in some cases, they may not welcome it.*" Yet, existing power grid is facing challenges related to efficiency, reliability, environmental impact, and sustainability. For instance, the low efficiency of current electric grid could lead to 8% of electric energy loss along its transmission lines, and the maximum generation capacity is in use only 5% of the time [14].

The emerging smart grid technology aspires to revolutionize traditional power grid with state-of-the-art information technologies in sensing, control, communications, data mining, and machine learning [6, 14]. Worldwide, significant research and development efforts and substantial investment are being committed to the necessary infrastructure to enable intelligent control of power systems, by installing advanced metering systems and establishing data communication networks throughout the grid. Consequently, power networks and data communication networks are envisioned to harmonize together to achieve highly efficient, flexible, and reliable power systems.

Among the various types of data transmitted over the smart grid, load curve data, which refers to the electric energy consumption periodically recorded by meters at points of interest across the power grid, has become the critical assets for utility companies to make right decisions on energy generation, billing, and smart grid operations. Load curve data, which "is beginning to give us a view of what the customer is actually experiencing, something that we've never ever seen before" [23], is precious user behavior data, and is an important type of big data.

Load curve data collected and reported from smart meters at end-users' premises is especially important for both energy supply and energy demand sides. On the demand side, it has direct impact on customers' energy bills and their trust on the still nascent smart grid technology. It can also provide important information for domotics [17, 26]. On the energy supply side, inaccurate load data may lead to large profit losses and wrong business decisions. In 2012, 126.8 million residential customers in the US used over $1,374$ billion kWh, which counts to over 33% of the total electric energy in the US [12]. The importance of this huge amount of energy and its financial implication cannot be over emphasized.

Nevertheless, it is ***unavoidable*** that load curves contain corrupted data and missing data, caused by various factors, such as malfunctioning meters, data packet losses in wireless networks, unexpected interruption or shutdown in electricity use, and unscheduled maintenance [5]. Due to the huge volume of load curve data, it is hard for utilities to manually identify corrupted load curve data. Unfortunately, problems caused by corrupted data are usually re-

alized only after it is too late, such as after a customer receiving a suspicious yet hard to rebut high energy bill.

As a concrete example, according to the news reports [20, 24], some customers in the province of British Columbia, Canada, were baffled by energy bills that are more than double what they were charged before the smart meter installation. While the problem could be identified by common sense and certain agreement might be reached by good faith negotiations [20, 24], fixing the questionable bill is another head-scratching and embarrassing issue to the utility. As a response to customer complaints, the utility normally took remedy actions, such as replacing the smart meters or taking back the smart meters for lab testing [24]. Such a remedy, however, can hardly be effective. According to CBC News [25], "Government estimates indicate there are about 60,000 smart meter holdouts (in the province)." Overall, the users and the utility company have the well-aligned interest and should work together to tackle this critical problem plaguing the electric power industry.

Techniques of load data cleansing have been proposed to deal with load data corruption problem recently [5]. Most existing load data cleansing methods are designed for the supply side (*i.e.*, electricity service providers), to help the utility companies find the corrupted data and protect their profits. From the supply side, the collected load data is usually aggregated data, *i.e.*, the energy consumption of a billing unit such as a house or a commercial building. When performing data cleansing on the supply side, due to the difficulty of obtaining extra knowledge behind the aggregated load data, most existing approaches apply outlier detection methods, *i.e.*, the data that deviates remarkably from the regular pattern is identified as corrupted data. Various assumptions about the data generation mechanism are required for outlier detection, but due to limited information, those assumptions are usually based on empirical knowledge or statistic features of the data. Such outlier detection methods are oblivious of appliances' various energy consumption models and may not be accurate or fair to customers. We call these methods *appliance-oblivious*. Such methods suffer from a few important deficiencies.

For example, the regression-based outlier detection methods find statistical patterns of load data and claim the data significantly deviating from the patterns as corrupted data. Nevertheless, such resulted outliers are not necessarily corrupted data. In addition, without the knowledge of appliances' energy consumption models, some "hidden" corrupted data is hard to detect. To be specific, the energy consumption of a group of appliances in a house or a building is a stochastic process. The stochastic feature makes it hard to establish a fixed pattern. Turning on/off any high-power appliance may lead to a steep change in load curve. Using appliance-oblivious data cleansing methods, the data generated under such a condition is likely to be captured as outliers.

As another example, appliance-oblivious methods cannot deal with "hidden" corrupted data. Fig. 1 shows an example of three appliances, A_1, A_2, and A_3, which have power ranges of $[2, 4]$, $[10, 12]$ and $[30, 32]$, respectively. The load data within some ranges such as $(4, 10)$, $(16, 30)$, and $(36, 40)$ cannot be generated by any combination of the three appliances. Nevertheless, such data may not be identified by existing outlier detection as corrupted data.

Recently, with the emergence of fine-grained in-house energy monitoring systems, customers now have the capability to monitor their own energy usage more closely and more accurately [30]. As such, users may possess more knowledge behind the data, *e.g.*, the decomposition of total energy consumption according to main appliances. Even if in-house energy monitoring system is not available, users should know the rated power of appliances', which are easily accessible from the appliances' manual, technical specifica-
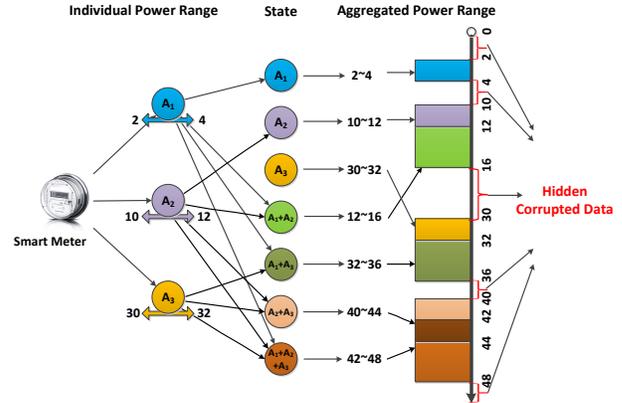
Figure 1: An example showing hidden corrupted data generated with three appliances

tion or public websites, such as [13]. This knowledge presents new opportunities to perform load data cleansing on the customer side directly or on the supply side with auxiliary information from customers. This new angle of tackling the corrupted load data problem can greatly improve the quality of load data.

In this paper, we tackle the practical problem of corrupted load curve data identification by developing an appliance-driven approach. Specifically, we make the following contributions:

- First, we define a new criterion in identification of corrupted load data. The criterion is aware of domain knowledge, including the power range of appliances and the physical laws behind valid load data.

- Second, we formally formulate the *Corrupted Data Identification Problem* (CDIP) and establish an optimization model to solve the problem. Furthermore, we introduce a new concept, called *virtual appliance*, in the objective function to help record corrupted data. Our empirical study in a proof-of-concept electricity usage test environment shows that a solution to the optimization problem is capable of precisely identifying the corrupted data, even without obtaining the exact on/off states of appliances. This nice feature indicates that our method is both effective and robust.

- We develop a *sequential local optimization algorithm* (SLOA) to approach CDIP efficiently. SLOA focuses on solving CDIP in a smaller time window, and considers the correlation between consecutive small windows. Our SLOA method offers an efficient heuristic solution and can achieve a very high detection precision. As an extra benefit, by applying the sequential optimization algorithm, we can easily handle consecutive corrupted data.

The rest of the paper is organized as follows. In Section 2, we review the related work. In Section 3, the corrupted data identification problem (CDIP) is formulated. To solve CDIP, an optimization model is developed in Section 4, and SLOA is applied to find an approximate solution in Section 5. We evaluate the performance of our method in Section 6 and test its robustness in Section 7. The assumptions is further discussed in Section 8. The paper is concluded in Section 9.

2. RELATED WORK

Most related literature treats the corrupted data the same as outliers in load pattern and focuses on outlier detection. A broad spectrum of techniques for outlier detection in load data have been developed, which include regression-based time series analysis, univariate statistical methods, and data mining techniques.

Regression-based time series analysis is the most widely used approach for outlier detection in load data [1, 5, 21, 22]. Mateos and Giannakis [22] developed a nonparametric regression method that approximates the regression function via ℓ_0-norm regularization. Chen *et al.* [5] proposed a nonparametric regression method based on B-spline and kernel smoothing to identify corrupted data. Abraham and Chuang [1] analyzed residual patterns from some regression models of time series and used the patterns to construct outlier indicators. They also proposed a four-step procedure for modeling time series in the presence of outliers. Greta *et al.* [21] considered the estimation and detection of outliers in time series generated by a Gaussian auto-regression moving average (ARMA) process, and showed that the estimation of additive outliers is related to the estimation of missing observations. ARMA is also utilized in [2, 1] as a fundamental model to identify outliers.

Univariate statistical methods are another type of techniques for outlier detection. Univariate statistical methods deal with outliers in load data by processing load data as one-dimensional real values [9, 15, 10]. Most univariate methods for outlier detection assume an underlying *a prior* distribution of data. The outlier detection problem is then translated to finding those observations that lie in the so-called outlier region of the assumed distribution, which is defined by a confidence coefficient value [10]. Since the statistical methods are susceptible to the number of exemplars, a simple but effective method named Boxplot or IQR is proposed in [31] to deal with small-sized exemplars.

In addition to the above methods, data mining techniques are also applied to identify outliers, such as k-nearest neighbor [29], k-means, k-medoids [4], and DBSCAN [18]. As a type of clustering methods, they group data with similar features, and identify data items that do not strongly belong to any cluster or far from other clusters as outliers. Recently, Aggarwal [3] provided a thorough survey on outlier detection.

Nevertheless, all the above methods do not consider the special physical laws behind the load data. Regression-based methods assume that the data follows a certain pattern, which can be modeled by a function governed by a set of parameters; univariate methods assume that the data is sampled from a certain known distribution; clustering methods assume that the data is well structured as clusters and the corrupted data deviates significantly from the normal structure. Obviously, the underlying assumptions in the existing methods are quite general and do not capture the specific features of load data well. Our paper fills the gap and differs from the existing literature by offering a completely new angle to address the load data corruption problem.

3. PROBLEM DEFINITION

In this section, we present a formal problem definition. Before that, we first describe an energy consumption model and discuss the generation rules of load data.

3.1 Energy Consumption Model

Load data is time series data that records users' energy consumption. It is collected by smart meters *periodically* at a certain sampling frequency. Without loss of generality, we assume that the time is slotted, with each timeslot equal to the sampling interval time. In the rest of the paper, we thus use the terms "*time*", "*timeslot*" and "*sampling interval*" interchangeably.

We denote the load data from timeslot $t = 1$ to timeslot $t = n$ in a column vector as

$$Y \equiv [y_1, y_2, \cdots, y_n]^T, \qquad (1)$$

where each value y_i in the vector represents the aggregated energy consumption of all appliances in a property, say a house at timeslot i. The energy consumption at a time instant depends on the appliances' on-off states and their individual power level.

We assume that a house includes m appliances in total, and the power of the k-th appliance is p_k (*watts*). At any time instant, if we record the power level of each individual appliance, we can define an m dimensional column *power vector* to capture energy consumption of the house:

$$P \equiv [p_1, p_2, \cdots, p_m]^T. \qquad (2)$$

Note that the power level of an appliance normally does not remain at a fixed value but changes in a certain range. For this reason, we define two m dimensional column vectors, denoted as P_l and P_u, respectively:

$$P_l = [l_1, l_2, \cdots, l_m]^T \qquad (3)$$
$$P_u = [u_1, u_2, \cdots, u_m]^T, \qquad (4)$$

where l_i and u_i represent the lower and upper bounds of the power level of the i-th appliance, respectively. A power vector P is called *valid* if, for each value p_i in P, $l_i \leq p_i \leq u_i$.

At any instant, the state of an appliance may be either *on* or *off*. We use an $n \times m$ 0-1 *state matrix*, $S = [S_{ij}]_{n \times m}$, to record the states of the m appliances from time $t = 1$ to $t = n$, where $S_{i,k} = 1$ indicates that the k-th appliance is on at time i, and 0 otherwise. In addition, we call the i-th row of S a *state vector* at time i, denoted by:

$$S_i \equiv [S_{i,1}, S_{i,2}, \cdots, S_{i,m}]. \qquad (5)$$

3.2 Generation Rules of Load Data

We have the following observations. First, a valid load data element y_i (in *watt-hours*) should be equal to the inner product of the state vector and the power vector at $t = i$, multiplied by the sampling interval time. This is a basic physical law for load curve data generation. Second, since the sampling interval is typically small, we assume that the probability that an appliance has more than one on-off switch events during a timeslot is negligible. In addition, the total number of on-off state switches of all appliances during a timeslot should be small. This feature is called the *temporal sparsity* of on-off switching events. Intuitively, this feature means that in normal operation it is unlikely that the household turns on/off many appliances in a short time. Based on the above observations, we define the generation rules of load data.

DEFINITION 1 (GENERATION RULES). *Assume that the initial state of appliances is S_0. We claim that each valid load data, y_i, must satisfy the following rules:*

$$\begin{cases} S_i \cdot P_l/f \leq y_i \leq S_i \cdot P_u/f \\ \|S_i - S_{i-1}\|_1 \leq \delta, \end{cases} \qquad (6)$$

where f is data sampling frequency, $1 \leq i \leq n$, and δ is the upper bound on the total number of on-off state switches for m appliances during a sampling interval.

Note that the energy consumption value (watt-hours) is calculated with power value (watt) multiplied by time $1/f$ (hour). To

keep our discussion simple, we assume that a valid initial state vector S_0 is given at this moment. We will relax this assumption later and show that the impact of an inaccurate initial state vector quickly becomes negligible as long as the system runs for just a little while (Section 7).

Based on the above generation rules, *corrupted data* is the values that break any of the rules.

DEFINITION 2   (CDIP). *The **corrupted data identification problem (CDIP)** is, given load data $Y = \{y_1, y_2, \cdots, y_n\}$, power bound vectors P_l, P_u, and a sampling frequency f, find corrupted data items that violate any of the generation rules, i.e., $C \equiv \{y_i : y_i$ violates (Equation 6), for $1 \le i \le n\}$.*

4. AN IMPORTANT STEP TOWARDS SOLVING CDIP

To solve CDIP, a naïve idea is to find all the solutions satisfying the constraints in (Equation 6), by brute-force, exhaustive search for all possible appliance states. This method is very time-consuming. Even for a small data set it is very costly to find the answer. Since the generation rules can be considered as constraints in an optimization problem, we will show how the problem can be transformed to an optimization problem, which sheds light on a fast solution to an approximate problem.

DEFINITION 3   (VIRTUAL APPLIANCE). *Besides the real appliances, we introduce a virtual appliance into the system. Its associated power is called virtual power, and we record the values of virtual power from time $t = 1$ to $t = n$ in a virtual power vector*

$$V \equiv [v_1, v_2, \cdots, v_n]^T, \tag{7}$$

where $v_i \in (-\infty, +\infty)$ denotes the virtual power at time $t = i$.

By introducing the virtual appliance, we can develop the following optimization model to solve CDIP:

$$
\begin{aligned}
\underset{S_i, v_i}{\text{minimize}} \quad & \|V\|_1 \\
\text{subject to} \quad & (S_i \cdot P_l + v_i)/f \le y_i \le (S_i \cdot P_u + v_i)/f \\
& \|S_i - S_{i-1}\|_1 \le \delta \\
& S_{i,j} \in \{0, 1\} \\
& 1 \le i \le n \\
& 1 \le j \le m
\end{aligned}
\tag{8}
$$

To understand the rationale behind the formulation of Equation (8), it is worthwhile to point out that $v_i \in V$ will come into play whenever S_i cannot satisfy the generation rules, *i.e.*, the virtual appliance is "turned on" when the load data y_i is corrupted. Thus, v_i essentially makes a record to the corrupted data. After obtaining the final solution to Equation (8), the v_i variables with non-zero values indicate the corrupted data, *i.e.*,

$$C = \{y_i : v_i \ne 0 \text{ for } 1 \le i \le n\}. \tag{9}$$

We try to minimize ℓ_1-norm, because it is proven that for most large under-determined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution (*i.e.*, resulting in the minimal number of non-zero values of v_i) [11]. In addition, a larger v_i value means that a corrupted y_i is farther away from a valid range. In this sense, v_i can be also regarded as the *corrupted degree* of y_i.

It can be proved that CDIP is NP-hard by reducing the Traveling Salesperson Problem (TSP) to CDIP. We omit the proof due to space limit. By investigating the special structure of the problem, however, we can develop an effective heuristic algorithm introduced in the next section.

Algorithm 1 Sequential Local Optimization Algorithm

Input: Load data $\{y_1, y_2, \cdots, y_n\}$, power bounds P_l, P_u, initial state S_0, sampling frequency f, local time window size w.
Output: Corrupted data set C, corrupted degree $v_i, 1 \le i \le n$
 1: $v_0 = 0$
 2: $C = \varnothing$
 3: **for** $k = 1 : n$ **do**
 4:     Solve Problem (Equation 10), and obtain v_i and S_i where $k \le i \le k + w - 1$
 5:     **if** $v_k \ne 0$ **then**
 6:         $C = C \cup \{y_k\}$
 7:         $S_k = S_{k-1}$
 8:     **end if**
 9: **end for**
10: **return** $C, \{v_1, v_2, \cdots, v_n\}$

---

5. SEQUENTIAL LOCAL OPTIMIZATION

In this section, we propose a *Sequential Local Optimization Algorithm* (SLOA) and develop a quantitative strategy to estimate the minimum local window size.

5.1 SLOA

The temporal sparsity of corrupted load data suggests that we can perform optimization in a smaller, local time window. By considering the correlation between consecutive timeslots, we design a Sequential Local Optimization Algorithm (SLOA). Without loss of generality, we take a load data from time $t = 1$ to $t = n$ as an example to show the major steps of SLOA.

Step 1. Consider a small time window with size of $w, 1 \le w < n$, which starts from time k to time $k + w - 1$. Given the state vector at time $k - 1$, *i.e.*, S_{k-1}, we consider the following optimization problem:

$$
\begin{aligned}
\underset{S_i, v_i}{\text{minimize}} \quad & \sum_{i=k}^{k+w-1} |v_i| \\
\text{subject to} \quad & (S_i \cdot P_l + v_i)/f \le y_i \le (S_i \cdot P_u + v_i)/f \\
& \|S_i - S_{i-1}\|_1 \le \delta \\
& S_{i,j} \in \{0, 1\} \\
& k \le i \le k + w - 1 \\
& 1 \le j \le m
\end{aligned}
\tag{10}
$$

By setting $w \ll n$, we can significantly reduce the search space. Actually, we can show that the computational complexity to solve the above problem is $O(M^w)$, where $M = \binom{m}{0} + \binom{m}{1} + \cdots + \binom{m}{\delta}$. Since m is the total number of appliances and $\delta \ll m$, the problem can be solved quickly using tools such as *CVX* 2.0 with a *Gurobi* engine [8].

Step 2. For the k-th time window that starts from time k, we use the following strategy to handle consecutive corrupted data: if the data point at time k is identified to be corrupted, *i.e.*, $v_k \ne 0$, recover the current state vector from the previous one, *i.e.*, set $S_k = S_{k-1}$.

Step 3. Repeat above two steps from $k = 1$ to $k = n$, and solve problems in form of Equation (10) sequentially. After n iterations, we can get a sequential solution $v_1, v_2, \cdots, v_n$. Thus, the corrupted data set is $C = \{y_i : v_i \ne 0, 1 \le i \le n\}$, in which v_i is the corrupted degree of load data y_i.

Algorithm 1 shows the pseudo code of SLOA.

5.2 Estimation of Local Window Size

Clearly, one key question in SLOA is to determine a suitable size of the local window. In principle, we want the size to be as small as possible to speed up the calculation, but a size too small may result in a poor solution largely deviating from the global optimal one. For example, in the extreme case of $w = 1$, SLOA becomes a simple greedy search algorithm, where the final solution may not be good. On the other hand, if $w = n$, the problem becomes the same as Equation (8), which is hard to solve. *What is the minimum local window size that leads to a nearly global optimal solution?*

Since it is hard to obtain a strict proof that the local optimal solutions together lead to the global optimal solution, we use the following heuristics to estimate the minimum local window size: within the local window, it should be possible that one state vector can be transited to any other state vectors in the vector space. In other words, within the local window, we should cover all possible state vectors in the search. This heuristics sheds light on finding the minimum local window size, as formulated below.

DEFINITION 4 (OVERLAP INDEX). *Consider m appliances denoted by a set $\{R_1, R_2, \cdots, R_m\}$, where $R_i \equiv [l_i, u_i] \subset \Re$ and represents the i-th appliance's power range. The overlap index of m appliances is defined as:*

$$O \equiv \frac{\sum_{i=1}^{m} \int_{p_{min}}^{p_{max}} I\left(R_i \cap \{x\}\right) dx}{\int_{P_{min}}^{P_{max}} I\left((\cup_{i=1}^{m} R_i) \cap \{x\}\right) dx} \quad (11)$$

where p_{max} and p_{min} stand for the maximum and minimum power of all appliances, respectively, and $I(x)$ is an indicator function

$$I(x) = \begin{cases} 1, & x \neq \varnothing \\ 0, & x = \varnothing \end{cases} \quad (12)$$

Note that the denominator $\int_{P_{min}}^{P_{max}} I((\cup_{i=1}^{m} R_i) \cap \{x\}) dx$ includes all valid power values, *i.e.*, the ones that can be covered by at least one appliance's power range. We can see that the overlap index represents the number of appliances whose power range covers a valid power value, averaged over the whole power range of all appliances. In particular, $O = 1$ indicates that no pair of appliances have overlapped power, and $O = m$ means that all appliances have the same power range. Intuitively, when O is large, there is a good chance of finding multiple local optimal solutions to Equation (10), since there are multiple equivalent choices to turn on/off appliances in each iteration.

With the heuristics in estimating the minimum local window size, we have the following result.

LEMMA 1. *Given the overlap index O of m appliances and the upper bound δ on the total number of on-off state switches in a timeslot, in order to get the nearly global optimal solution to Equation (8) via Equation (10), the minimum local window size $w = max\{\lceil \frac{m}{\delta \cdot O} \rceil, 1\}$.*

PROOF. We prove the following condition holds: within the local window, we can cover all possible state vectors in the search.

First, the value of w relates to upper bound $\delta (\leq m)$ on the total number of on-off switches in one timeslot. It is obvious that from time $t = i$ to $t = i + 1$, the state vector S_i can only change to another state vector S_{i+1}, with $\|S_{i+1} - S_i\|_1 \leq \delta$. If $\delta = m$, then within one step, a state vector is allowed to change to any other state vector. On the other hand, if $\delta = 1$, within one step, a state vector can only change one value in the vector. In other words, from one state vector, it requires at least $\lceil \frac{m}{\delta} \rceil$ timeslots to reach any other state vector in the state vector space, *i.e.*, $w \geq \lceil \frac{m}{\delta} \rceil$.
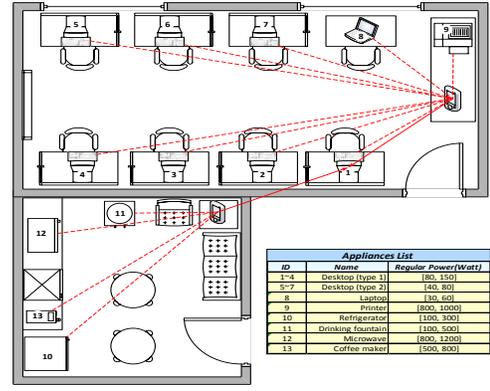
Figure 2: Energy monitoring platform and appliances' power ranges

Second, the overlap index O can reduce the value of w. Based on the meaning of O, m appliances with overlap index O are equivalent to $\lceil m/O \rceil$ appliances without overlapped power. Replace m with m/O, we can get $w \geq \lceil \frac{m}{\delta \cdot O} \rceil$. Considering $w \geq 1$, we conclude:

$$w \geq max\{\lceil \frac{m}{\delta \cdot O} \rceil, 1\}. \quad (13)$$

Since we want w to be as small as possible, $w = max\{\lceil \frac{m}{\delta \cdot O} \rceil, 1\}$. □

Note that, although the minimum local window size obtained above is an estimation, it works effectively in our experiments with real-world data as well as with synthetic data.

5.3 Algorithm Analysis

Given n load values, m appliances, and the upper bound $\delta (\leq m)$ on the total number of on-off state switches in a timeslot, the computational complexity of the original problem (Equation 8) is $O(M^n)$, where $M = \binom{m}{0} + \binom{m}{1} + \cdots + \binom{m}{\delta}$. Using SLOA, solving the optimization problem (Equation 10) for n times results in the time complexity of $O(n \cdot M^w)$, where $w \in Z^+$ and $w \ll n$. Please note that the appliance number m is a constant value and w is also a small constant.

Obviously, the larger the value of w, the higher the computational complexity. Fortunately, the overlap index of appliances in a house/building is usually high, as observed in our real-world testbed. This fact allows us to select a small local window size following Lemma 1. Therefore, in the application scenarios, SLOA can approach the NP-hard problem heuristically and efficiently. We will show that this algorithm indeed can provide a good solution with abundant experimental results in the following sections.

6. EXPERIMENTAL EVALUATIONS

6.1 Evaluation on Real Data

6.1.1 Case 1: Evaluation for Laboratory Facility

We evaluate our method with real-world trace data from a real-world energy monitoring platform. We monitored the appliances' energy consumption of a typical laboratory and a lounge room on the fifth floor of the Engineering/Computer Science Building at University of Victoria. The real-time power of laptops, desktops and some household appliances was recorded. Each appliance's power level was measured every 10 seconds and the measurement results were transmitted with ZigBee radio to a server that stores

Table 1: Results of corrupted data identification on university facility data: appliance-driven approach vs. B-spline smoothing

	appliance-driven approach				B-spline Smoothing			
	$w=1$	$w=2$	$w=3$	$w=5$	$df=128$	$df=188$	$df=258$	$df=388$
Precision	89.29%	95.83%	85.29%	84.38%	48.68%	50.00%	51.39%	47.44%
Recall	50.00%	46.00%	58.00%	54.00%	72.55%	74.51%	82.35%	72.55%
$F-measure$	64.10%	62.16%	69.05%	65.85%	58.27%	59.84%	60.16%	57.36%

the data. The monitored appliances and their regular power[1] are shown in Fig. 2.

We collect the data over a two-month period. Fig. 3 demonstrates one-week and one-day load data collected by our platform. For clear illustration, we only show one-day data as an example. Note that even in a lab setting like ours, there indeed exists some apparent corrupted data, indicated by the dashed red dots in Fig. 3[2].
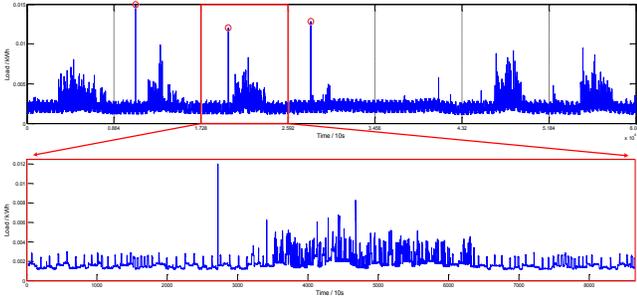
Figure 3: One-week and one-day load data collected via the energy monitoring platform.

In order to introduce more corrupted data, we asked three students to distort the load data with "falsification", *i.e.*, they were asked to arbitrarily modify the aggregated load data within the range of $[0,\infty)$. These changed data together with the original corrupted ones were labeled and used as the ground-truth to verify the performance of our method.

Since the existing appliance-oblivious load data cleansing methods, such as B-spline smoothing, detect outliers and consider outliers as corrupted data, we use the terms "outliers" and "corrupted data" interchangeably hereafter. For outlier detection, four statistical results can be obtained: (1) true positive (TP), the number of points that are identified correctly as outliers; (2) false positive (FP), the number of points that are normal but are identified as outliers; (3) true negative (TN), the number of points that are normal and are not identified as outliers; (4) false negative (FN), the number of points that are outliers but are not identified. Using TP, FP, TN and FN, we evaluate the following three broadly-used performance metrics: precision, recall, and F-measure. Precision is the ratio of the number of correctly detected corrupted values over the total number of detected values; recall is the ratio of the number of correctly detected values over the number of pre-labeled corrupted values; and the F-measure is a harmonic mean of precision and recall, *i.e.*,

$$F\text{-}measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}. \tag{14}$$

For comparison, we implement and test an appliance-oblivious data cleansing method, B-spline smoothing, which is introduced in [5] to identify corrupted load data. In the B-spline smoothing

method, we set the confidence coefficient $\alpha = 0.05$, which results in a confidence interval of 95%. We treat the degree of freedom (df) as a variable, whose value is trained when smoothing the load curve data. For our method, the *overlap index* is obtained as $O \approx 2$, and the upper bound of on-off switching events of appliances within the sampling interval is set to 2, *i.e.*, $\delta = 2$. According to Equation (13), the local window size, *i.e.*, the value of w in Algorithm 1, is set to 3. Since the value of local window size is an estimation, in order to obtain more comprehensive performance evaluation for our method, we also vary the local window size in a range.

Table 1 summarizes some of the results from the two methods. Furthermore, Fig. 4 and Fig. 5 illustrate one of the outcomes from our appliance-driven method and the B-spline smoothing method, respectively. From the results, we have the following interesting observations.

- Comparing to B-spline smoothing, our method performs much better in *Precision*, but worse in *Recall*. This shows that our method can identify the corrupted data more accurately, even though our output does not cover the completed set of all corrupted data. In addition, our appliance-driven method achieves a higher *F-measure*. *F-measure* reflects a balanced mean between precision and recall, indicating that our method has overall better performance.

- The performance of our method remains roughly the same when the local window size is beyond the minimum value estimated using Lemma 1. Further increase of local window size does not bring clear performance gain but takes longer running time. This suggests that our previous estimation on the minimum local window size for SLOA is appropriate.

6.1.2 Case 2: Evaluation for Households

In addition to the data collected from university facilities, we also evaluate our method with real-world trace data from some typical households. Load data of four houses (each house possesses $20 \sim 40$ appliances) were collected by Belkin Inc. and made on Kaggle [16]. With the data from one of the four households, we evaluate the performance of our appliance-driven method. The experimental process is similar to that in Case 1, and the results are summarized in Table 2.

Table 2: Results of corrupted data identification on household data ($w = 1$)

	Appliance-driven		B-spline Smoothing	
	$\delta = 2$	$\delta = 3$	$df = 188$	$df = 258$
Precision	67.86%	80.30%	36.84%	40.67%
Recall	73.79%	53.00%	70.00%	61.00%
$F-measure$	70.70%	63.86%	48.28%	48.80%

6.2 Evaluation on Large-Scale Synthetic Data

To thoroughly test our method, we evaluate its performance using large-scale synthetic data that simulates a large number of appliances and much diverse energy patterns. With different synthetic datasets, we can also test the robustness of our method.

---

[1] An appliance's regular power is an approximate range around the rated power where this appliance works.

[2] The corrupted data mainly comes from some incorrect power values from the laptop that occasionally reports impossible values such as hundreds of Watts.
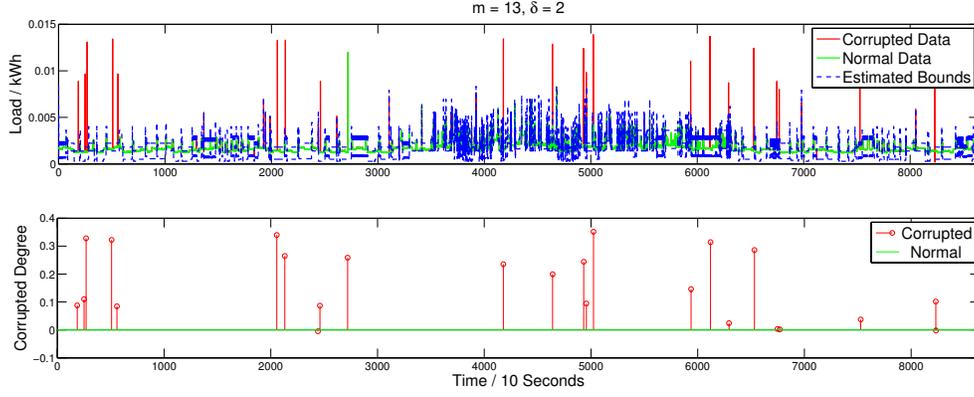
Figure 4: Result of corrupted data identification on university facility data with our appliance-driven method ($w = 1, \delta = 2$); **Estimated bounds denote the upper and lower power bounds based on current state vector; Corrupted degree indicates the value of the virtual appliance (Section 4.)**
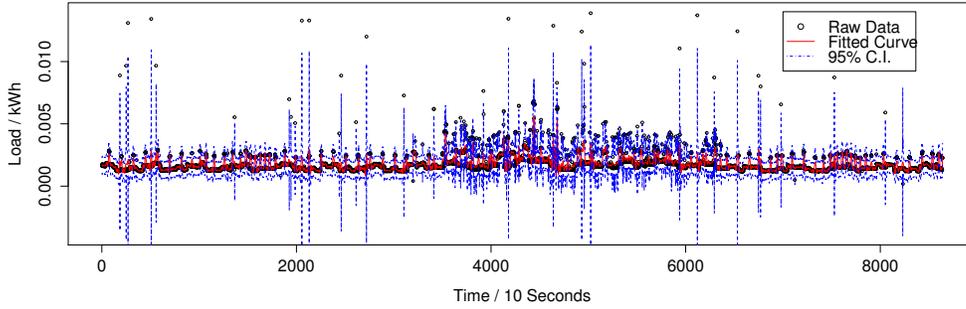
Figure 5: Result of corrupted data identification on university facility data with the B-spline smoothing method ($df = 258$)

6.2.1 *Load Data Generation via Monte Carlo Simulation*

There is no standard model for the load curve data of a house, since the data actually results from a complex process related to human activities. We thus use the *Monte Carlo* simulation to generate the load data using the following method:

- Given the lowest appliance power (P_{min}) and the highest appliance power (P_{max}), the lower bound of an appliance (p_l) is a random variable uniformly distributed between P_{min} and P_{max}. The upper bound of the appliance (p_u) is determined by a parameter called *power range ratio* (r) and is calculated by $p_u = \min\{p_l + random([0, rp_l]), P_{max}\}$, where $random([0, rp_l])$ returns a random number uniformly distributed in the range $[0, rp_l]$.

- At a given sampling frequency, each appliance reports its current power value, which is a random number uniformly distributed between the appliance's lower power bound and upper power bound. It reports 0 if its state is *off*.

- In a sampling interval, the number of total on-off switch events follows a Poisson distribution[3] with parameter λ.

---

[3]Poisson distribution is a good model for situations where the total number of items is large and the probability that each individual item changes its state is small. It has been broadly adopted to simulate events related to human behavior, such as the number of telephone calls in a telephone system.

- At the end of each sampling interval, the load data of the house is recorded as the aggregated power value of all appliances (i.e., the sum of all appliances' load values).

To introduce some corrupted data and test the effectiveness of our method, we "corrupt" some data values by replacing them with random values uniformly distributed between $[0, Max]$, where Max is a given large constant. The time interval of introducing corrupted data is assumed to follow an exponential distribution with the mean value of μ.

6.2.2 *Corrupted Data Identification on Large-Scale Appliances*

The parameters used to generate the synthetic data and the corrupted data are listed in Table 3.

Table 3: Parameter settings for load data generation and corruption

Parameter	Setting
Number of Appliances (m)	50
Sampling Frequency(f)	$1/6Hz$
Total Time Span	$3600s$
Lowest Appliance Power(P_{min})	$50w$
Highest Appliance Power(P_{max})	$2000w$
Power Range Ratio(r)	15%
Initial State(S)	$[0, 0, \cdots, 0]^T$
Poisson Parameter(λ)	5
Exponential Parameter(μ)	30
Corrupted Data Range	$[0, 50kW]$

Table 4: Results of corrupted data identification on synthetic data: appliance-driven method vs. B-spline smoothing

	appliance-driven Method			B-spline Smoothing			
	$\delta = 4$	$\delta = 5$	$\delta = 6$	$df = 140$	$df = 160$	$df = 180$	$df = 200$
Precision	93.94%	93.94%	100%	78.57%	86.49%	84.61%	84.21%
Recall	81.58%	81.58%	63.16%	86.84%	84.21%	86.84%	84.21%
$F - measure$	87.32%	87.32%	77.42%	82.50%	85.33%	85.71%	84.21%
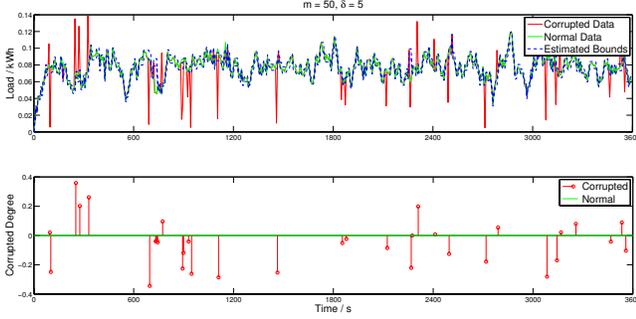
Figure 6: Result of corrupted data identification on synthetic data with our appliance-driven method ($w = 1, \delta = 5$)
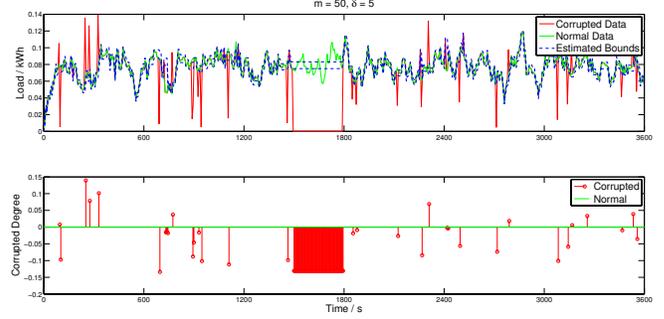
Figure 8: Identification of consecutive corrupted data with our appliance-driven method
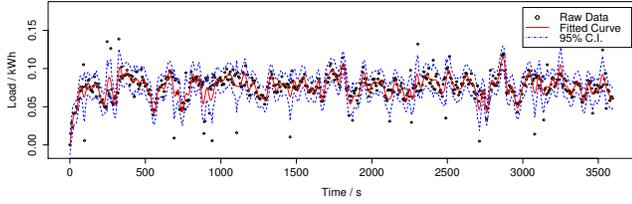
Figure 7: Result of corrupted data identification on synthetic data with B-spline smoothing method($df = 160$)
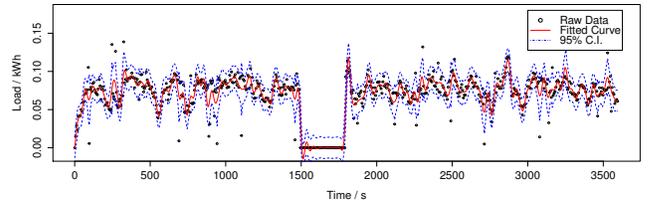
Figure 9: Identification of consecutive corrupted data with B-spline smoothing method ($df = 100$)

We treat the bound on the total number of on-off switches in a sampling interval δ as a variable. To speed up the processing, we set the local window size to 1. The small local window size may not lead to the best performance of SLOA. However, as shown in our experimental results, even with this setting, our method already performs better than B-spline smoothing. For the B-spline smoothing method, the degree of freedom (df) is set as a variable and is trained when smoothing the synthetic data.

The performance results of our method and the B-spline smoothing method are summarized in Table 4. Fig. 6 and Fig. 7 illustrate one of the outcomes from our method and the B-spline smoothing method, respectively.

From the results, we can see that the our method works effectively on large-scale synthetic data. In particular, the *precision* of our method increases with increase of δ, and can even reach 100%. This result indicates that our method can provide exactly correct identification when δ is large enough. Regarding the overall performance in view of *F-measure*, our method works better with a smaller δ value and outperforms B-spline smoothing clearly.

6.2.3 Identification of Consecutive Corrupted Data

In practice, we often meet the situation that all data within a small time interval are corrupted or lost. Consecutive corrupted data poses a big challenge to regression-based methods, as will be illustrated in this subsection.

To introduce consecutive corrupted data, we replace the load data in a small time window as 0s, as shown in the upper part of Fig. 8. We then use our method and the B-spline smoothing method to test the data. Fig. 8 and Fig. 9 illustrate one outcome from our appliance-driven approach and the B-spline smoothing method, respectively.

From the results, we can see that our method does much better than B-spline smoothing for consecutive corrupted data identification. With $\delta = 5$, our method can correctly identify all the corrupted data. On the other hand, even though we regulate the parameters for B-spline smoothing, it almost failed every time to identify even half of the corrupted data.

An interesting phenomenon can be found around the consecutive corrupted data in Fig. 9. There is an apparent trend with B-spline smoothing to fit the corrupted data. This is mainly because the B-spline smoothing method tries to fit the curve pattern and reduce the total bias error with global optimization, indicating that it cannot deal with consecutive corrupted data well.

7. ROBUSTNESS TESTING

One may question whether the performance of SLOA relies on a correct initial state vector, accurate information regarding appliances power ranges, and an accurate estimation on appliances' on-off states. all of such information may be hard to obtain in practice.

Table 5: Robustness tests with incorrect power ranges of appliances

	Widen (5%)		Widen (10%)		Shift& Widen (5%)		Shift& Widen (10%)	
	$\delta = 3$	$\delta = 4$	$\delta = 3$	$\delta = 4$	$\delta = 3$	$\delta = 4$	$\delta = 3$	$\delta = 4$
$Precision$	93.55%	87.50%	91.30%	94.12%	67.39%	90.91%	71.11%	84.62%
$Recall$	76.32%	55.26%	55.26%	42.11%	81.58%	78.95%	84.21%	57.89%
$F-measure$	84.06%	67.74%	68.84%	58.18%	73.81%	84.51%	77.11%	68.75%

To answer this question, we test the robustness of SLOA. We use the synthetic data created using the same parameters in Table 3. We first disclose the test results and then explain the reasons.

7.1 Impact of the Initial State

For this test, we change the initial state of an appliance to a random 0-1 value, and perform multiple tests. Fig. 10 shows one of the outcomes. We find that, even with an incorrect initial state, our method can always recover to correct load data after a few steps. This result indicates that our SLOA method is robust against inaccurate initial power state setting.
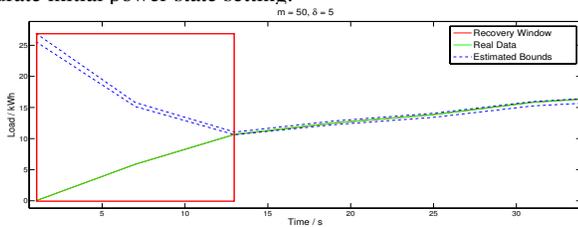
Figure 10: Fast recovery of estimated load starting from a random initial state

7.2 Impact of Power Ranges

In practice, we may not precisely know the power ranges of appliances. Based on this consideration, we run extra simulations to test the robustness of our method when the power ranges of appliances is inaccurate. We carry out two kinds of tests as follow.

- Widen power range: each appliance's power range is widened by 5% or 10%, respectively, with the center power value, *i.e.*, (upper bound − lower bound)/2, unchanged.

- Shift& widen power range: each appliance's lower power bound is increased by 5% or 10%, and upper bound is increased by 5% or 10%, respectively. Note that the above operations will shift the center power value as well as widen the power range.

We do not consider the situation where appliances' power ranges are narrowed, since intuitively a user can always widen an appliance's power range if she/he is not sure about the right values. The test results are summarized in Table 5. The results clearly indicate that, with inaccurate or even wrong power ranges of appliances, our method can still identify corrupted data with high precision.

7.3 Impact of State Vector

We have seen that our method can give correct bounds for energy consumption most of the time. Accordingly, we might infer that the estimated states of the appliances should be the same with the real situation, or at least quite close.

In order to verify this conjecture, we calculate the difference (one-norm distance) between the estimated state S_e and real state S_r at each time instance. Fig. 11 shows the result.

To our surprise, the estimated states are not close to the real states, and actually deviate remarkably from the real states. We can
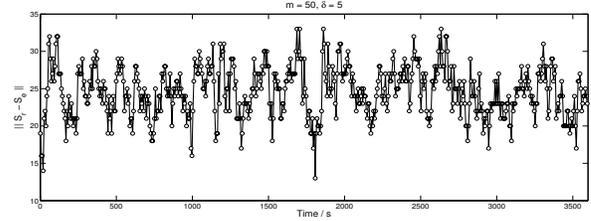
Figure 11: Difference between estimated state S_e and real state S_r

see that in Fig. 11, the mean distance between S_e and S_r is around 25, indicating that nearly half of the appliances are not estimated with correct states. This shows that the solution to the CDIP problem is not unique but multiple, and our method can provide right load data without need to always find the right states of appliances.

7.4 Why Is SLOA Robust?

In real life, a lot of appliances are with similar or overlapped power range. In this sense, we indeed do not need to know the exact state for similar appliances as long as we can give a good approximation for their total consumption. In addition, due to the temporal sparsity of on-off switch events in the short sampling interval and the fact that only some appliances are on at any time instant, the negative impact of inaccurate power range estimation on one appliance can be offset by the negative impact of incorrect state estimation of another appliance. The offsetting is enforced automatically by the optimization objective function that minimizes the gap between the actual load data and the estimated value.

8. FURTHER DISCUSSION

Our approach is based on the assumption that customers are willing to collaborate and provide their appliances' information. This naturally causes privacy concerns. Nevertheless, this assumption is nothing special in real-world applications, since customers have to put certain trust on service providers. Commercial energy monitoring platforms, such as PlotWatt [27] and PlugWise [28], indeed require users' appliance information.

Another concern of our solution is that users may not be able to know the power model of each appliance. Nevertheless, the information needed in our solution is simple and could be found from users' manual, technical specification or public websites such as [13, 19]. In addition, cheap per-appliance monitoring devices [7] can be used to obtain the required information. In our above test, it can be seen that it is unnecessary to precisely know the power ranges of appliances, and our solution can tolerate up to 20% of estimation errors in appliances' power ranges.

9. CONCLUSION AND FUTURE WORK

To answer the industrial call for improving quality of load data, we developed a new appliance-driven approach for corrupted data identification that particularly takes advantage of information available on the demand side. Our appliance-driven approach considers

the operating ranges of appliances that are readily available from users' manual, technical specification, or public websites [13]. It identifies corrupted data by solving a carefully-designed optimization problem. To solve the problem efficiently, we developed a sequential local optimization algorithm (SLOA) that practically approach the original NP-hard problem approximately by solving an optimization problem in polynomial time. We evaluated our method using real trace data from a university facility and four typical households, and large-scale synthetic data generated by Monte Carlo simulation. Test results indicate that our method can precisely capture corrupted data. In addition, SLOA is robust under various test scenarios, and its performance is resilient to inaccurate power range information or inaccurate power state estimation.

Our method greatly augments the arsenal of existing load data cleansing tools to minimize human effort in identifying corrupted data. How to replace aberrant values and missing values is out of our focus, because this issue is relevant to utilities' internal rules and thus requires human interaction. Considering various policies and methods for load data imputation will be our future work.

10. ACKNOWLEDGMENTS

11. REFERENCES

[1] B. Abraham and A. Chuang. Outlier detection and time series modeling. *Technometrics*, 31(2):241–248, 1989.

[2] B. Abraham and N. Yatawara. A score test for detection of time series outliers. *Journal of time series analysis*, 9(2):109–119, 1988.

[3] C. C. Aggarwal. *Outlier Analysis*. Springer, 2013.

[4] R. J. Bolton, D. J. Hand, et al. Unsupervised profiling methods for fraud detection. *Credit Scoring and Credit Control VII*, pages 235–255, 2001.

[5] J. Chen, W. Li, A. Lau, J. Cao, and K. Wang. Automated load curve data cleansing in power systems. *IEEE Transactions on Smart Grid*, 1(2):213–221, 2010.

[6] S.-y. Chen, S.-f. Song, L. Li, and J. Shen. Survey on smart grid technology. *Power System Technology*, 33(8):1–7, 2009.

[7] C. Cost. Current cost products - envir. http://www.currentcost.com/product-envir.html/, accessed in Feb. 2014.

[8] CVX. Matlab software for disciplined convex programming. cvxr.com/cvx, accessed in Jul. 2013.

[9] H. David. Robust estimation in the presence of outliers. *Robustness in statistics*, 1:61–74, 1979.

[10] L. Davies and U. Gather. The identification of multiple outliers. *Journal of the American Statistical Association*, 88(423):782–792, 1993.

[11] D. L. Donoho. For most large underdetermined systems of linear equations the minimal l_1 norm solution is also the sparsest solution.

[12] EIA. Electric power annual 2012. http://www.eia.gov/electricity/annual/?src=Electricity-f4, accessed in Jan. 2014.

[13] EPA. A tool of home product finder. http://www.energystar.gov/productfinder/, accessed in Dec. 2013.

[14] H. Farhangi. The path of the smart grid. *Power and Energy Magazine, IEEE*, 8(1):18–28, 2010.

[15] U. Gather. Testing for multisource contamination in location/scale families. *Communications in Statistics-Theory and Methods*, 18(1):1–34, 1989.

[16] K. Inc. Belkin energy disaggregation competition. https://www.kaggle.com/c/belkin-energy-disaggregation-competition, accessed in Jul. 2013.

[17] M. Jahn, M. Jentsch, C. R. Prause, F. Pramudianto, A. Al-Akkad, and R. Reiners. The energy aware smart home. In *Future Information Technology (FutureTech), 2010 5th International Conference on*, pages 1–8. IEEE, 2010.

[18] H.-P. Kriegel and M. Pfeifle. Density-based clustering of uncertain data. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 672–677. ACM, 2005.

[19] L. Laboratory. Standby power summary table. http://standby.lbl.gov/summary-table.html, accessed in Feb. 2014.

[20] J. Lang. A bad case of smart meter regret. http://www.cloverdalereporter.com/news/189923521.html, accessed in Jan. 2014.

[21] G. M. Ljung. On outlier detection in time series. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 559–567, 1993.

[22] G. Mateos and G. B. Giannakis. Robust nonparametric regression via sparsity control with application to load curve data cleansing. *IEEE Transactions on Signal Processing*, 60(4):1571–1584, 2012.

[23] J. McMahon. Big data from smart grid tells utilities more than they want to know. *Forbes*, September 2013.

[24] S. Moneo. Bc hydro pulls smart meters for testing. http://www.theglobeandmail.com/news/british-columbia/bc-hydro-pulls-smart-meters-for-testing/article534384/, accessed in Jan. 2014.

[25] C. News. BC Hydro outlines smart meter refusal cost. September 13, 2013.

[26] T. A. Nguyen and M. Aiello. Energy intelligent buildings based on user activity: A survey. *Energy and buildings*, 56:244–257, 2013.

[27] PlotWatt. https://plotwatt.com/, accessed in Jan. 2014.

[28] PlugWise. http://www.plugwise.com/, accessed in Jan. 2014.

[29] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. In *ACM SIGMOD Record*, pages 427–438. ACM, 2000.

[30] J. Stragier, L. Hauttekeete, and L. De Marez. Introducing smart grids in residential contexts: Consumers' perception of smart household appliances. In *Innovative Technologies for an Efficient and Reliable Electricity Supply (CITRES), 2010 IEEE Conference on*, pages 135–142, 2010.

[31] J. W. Tukey. Exploratory data analysis. *Reading, MA*, 231, 1977.

Communications on pure and applied mathematics, 59(6):797–829, 2006.