©IEEE, 2020. This is the author's version of the work. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purpose or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the copyright holder. The definite version will be published at 2020 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT).

Increasing the Accuracy and Speed of Universal Non-Intrusive Load Monitoring (UNILM) Using a Novel Real-Time Steady-State Block Filter

Richard Jones, Alejandro Rodriguez-Silva, Stephen Makonin
School of Engineering Science, *Simon Fraser University*, Burnaby, Canada
Email: rtj4@sfu.ca, ara61@sfu.ca, smakonin@sfu.ca

***Abstract*—Non-intrusive load monitoring (NILM) is a research field focused on developing algorithms that can accurately track constituent electrical loads in a system using only the aggregate signal alone (i.e., smart meter). It is widely understood that having a clean signal free of noise and transient behaviour, whether for event-based or state-based methods, can lead to more accurate solutions that will eventually solve the NILM problem. We propose a fast and highly reliable method for producing a block-like representation of signals. Using the same data and disaggregation technique, we compare our algorithm with a recent similar effort and show significant improvements in accuracy (98% vs. 94% tracked energy over three appliances) and run-time (143ms vs. 891s). Application of our method to raw mains power data shows it can generalize to more complex cases.**

***Index Terms*—unsupervised learning, disaggregation, non-intrusive load monitoring, NILM, universal NILM, UNILM, adaptive filter, smart meter, smart grid**

I. Introduction

Non-intrusive load monitoring (NILM) [1] infers or tracks constituent electrical loads by only using the aggregate power signal (i.e., smart meter) – often referred to as disaggregation. Disaggregation of constituent loads in a system is a difficult and complex problem, owing to the fact that electrical loads in a home often significantly overlap in time or are even concurrent with one another [2]. Many statistical models and algorithms have been developed to solve this problem, and significant results have been achieved in the supervised-learning context, where algorithms are fed labelled data from individual loads in order to learn characterizing features. Data can be taken from different publicly available datasets [3] to learn these features. However, generalization from these learned models to new homes with different appliances has proven difficult [4]. Unsupervised techniques have consequently come to the forefront of research in NILM, where only aggregate data can be used, forcing the development of algorithms able to learn immediately upon installation in a new home.

Regardless of method, it is often the case that noise and short-lived appliance behaviour are prohibitive to the success of NILM algorithms. The intent of this paper is illustrated by what we consider to be the contributions of our work to the NILM field:

Research supported by NSERC Discovery Grant RGPIN-2018-06192.

A. Research Contributions

The present work provides the several contributions to the NILM field:

1) We demonstrate an algorithm (or filter) capable of producing a clean, block-like representation of aggregate data using a conceptually simple event-detection method.
2) We show significant improvements in run-time and accuracy relative to a recently published effort [5].
3) We demonstrate the generalization of this method to real use cases in complex aggregate examples.

II. Background

Several disaggregation methods in NILM – both supervised and unsupervised – either require or benefit from a clean aggregate and/or sub-metered data with well-defined power consumption levels separated by 'on' or 'off' events. In other words, a block-like representation of the data, with a single power level representing each combination of appliance states in the home. Super-state hidden Markov models (SSHMMs) introduced in [6] are one example of a method benefiting from such a representation. In these models, all possible combinations of appliance states are enumerated (called super-states) and using the training data, probabilities are associated with the emission of a particular power draw by a given super-state, and also with the likelihood of transition from one super-state to another. Because the number of super-states increases rapidly with the number of loads in the house, the Gaussian distributions on emission and transition built from the training data become congested, and the confidence with which the model can ascribe a given power draw to a single super-state often decays quickly. Noise in the training data or the raw aggregate can very easily cause a misclassification during disaggregation. If instead a block-like representation of the data was passed to a SSHMM, the transition and emission distributions would be much more localized, allowing the model to tolerate a higher number of loads.

Another example of the benefit of a block-like representation is the event-based disaggregation method introduced by Dong and collaborators in 2013 [7]. In their work, they suggested modelling appliances as finite-impulse response (FIR) filters, whose response to piece-wise constant inputs (i.e. on/off inputs to the filter modelling each appliance) could reconstruct the observed signals in the aggregate. Following the notation laid out in their work, for training a given load i, we know in the training data the on/off state $u_i^z[t] = \{0, 1\}$ of the load for any time t, as well as the resulting power draw, $z_i[t]$. In a standard channel estimation context, the n_i weights needed for modeling the response are determined by least-squares optimization and stored in a vector $\beta_i = [b_{i,1}\ b_{i,2}\ ...\ b_{i,n_i}]^\top$. In other words, the training set of each load can be recreated by

$$y_i^z[t] = \sum_{j=1}^{n_i} \beta_i^\top u_i^z[t-j] + e_i[t], \tag{1}$$

where $e_i[t]$ is a noise/error term. During testing, the aggregate is known and recovering the power consumption of each load is needed:

$$y[t] = \sum_{i=1}^{D} y_i[t] = \sum_{i=1}^{D} \beta_i^\top \psi_i[t] + e_i[t], \tag{2}$$

where

$$\beta_i^\top \psi_i = \sum_{j=1}^{n_i} b_{i,j} u_i[t-j]. \tag{3}$$

And so the disaggregation problem can be stated as finding the $u_i[t]$ at each t for each of the $i = 1 : D$ loads given only the aggregate sum, $y[t]$; no doubt a difficult problem. They attempted to solve this problem by arranging the on/off state of each appliance to be disaggregated in a vector, $u[t] = [u_1[t]\ u_2[t]\ ...\ u_D[t]]$, which is constant between events (i.e. where one or more loads change state). If noise were removed (and assuming no unmetered events take place), the power draw between events k and $k+1$ would be constant, which we can call the steady state power draw for segment k. When an appliance turns on, we can examine the jump in the steady state from one segment to the next and estimate what change in the on/off vector Δu occurred. This change can be estimated for a given time-step based on how each filter (and every combination thereof) responds to a step impulse, also called the zero-state response, ξ_i of the i^{th} filter. This zero-state response models the transient of a given appliance as it approaches its steady state value.

Assuming a change did occur in the aggregate, we can therefore assign that event to the appropriate load or loads by considering

$$\Delta u = \text{argmin}_{\Delta u} \left|\left| y[t_k : t_{k+1}] - \Big(y_{ss}(k) + \xi(\Delta u) \Big) \right|\right|_2^2, \tag{4}$$

where $y[t_k : t_{k+1}]$ is the observed signal in the k^{th} segment, $y_{ss}(k)$ is the steady state signal in the k^{th} segment, and $\xi(\Delta u) = \sum_{i=1}^{M} \xi_i$, i.e. $\xi(\Delta u)$ is the summed zero-state responses for each of the M filters that change in the argument, Δu. This y_{ss} value, taken over all segments k, is exactly the block-like representation of the aggregate signal mentioned previously. In their work, however, Dong et al. provide no mention of how to produce this representation, saying instead that it can be estimated from the data and is assumed to be known.

A recent attempt at producing this block-like representation was made by [5], the motivation for which was an unsupervised/semi-supervised knapsack optimization method. In their work, classification was achieved by characterizing all load states via Gaussian distributions on their power demand (or y_{ss} value) and the duration of these states. In other words, the well-defined power draw and duration of a given event observed in the aggregate can be assigned with a certain probability to an appropriate load, based on the difference in its demand and duration values. Once the loads were separated, they proposed a labelling structure whereby regionally accurate distributions on demand and duration are obtained from data set surveys in the NILM field and/or data sheet surveys from manufacturers of particular appliance types. A kind of two-dimensional map, shown in [5], is helpful to illustrate the labelling process for North American appliances. When the loads are all well-separated in this two-dimensional demand/duration representation, this technique is effective. It's inevitable, however, that as more loads and individual load states are added to this 2D mapping, overlap will occur and disaggregation performance will decline.

Nevertheless, they produced the block-like representation of the aggregate by passing raw data through a pipeline of filters consisting of five steps shown in table I. Although quite promising results were achieved, the filters necessary to produce the desired result were prohibitively demanding in terms of time complexity, needing nearly 15 minutes to pre-process and filter 5400 samples. Our proposed method shows significant improvement in run-time, and demonstrable consistency even in aggregate data, which proved difficult for the filter pipeline [5].

III. METHODOLOGY

The premise of our proposed method is extremely simple: find events in a window of the aggregate, and replace the intermediate values with the mean value over each segment in the window. This method necessarily excludes real-time disaggregation as a possibility, but few use cases in NILM require user updates at the sampling rate (1 Hz for modern smart meters). Providing 'down-sampled' user updates at say 30 s or 1 min intervals is often more than sufficient. This assumption is made in the current work, and allows events to be found easily, rather than estimated at every sample.

For an event to justify disaggregation, the change in power observed in the aggregate is generally large relative to the noise. Small loads have negligible impact on the overall energy consumption and are often ignored. For this reason, examining the differential of the raw aggregate allows us to find all events larger than some desired threshold. Since noise increases with overall power draw, as well as for certain loads or load states, a wavelet denoising approach allows a first-pass reduction in noise with little cost to overall run-time. Because wavelets are well-defined in time and frequency, we can maintain sharp events while removing sub-threshold noise components. In fact, run-time tests show that an initial wavelet denoising actually improves overall run-time, since there are far fewer events in the aggregate to consider, and also less post-processing to be done on the output as a result. In the present work, a Daubechies wavelet was used, with a single vanishing moment, a universal threshold denoising method, and a level independent noise estimate.

Once events are determined, the intermediate values are smoothed using a moving median filter. Since it was already determined that no appreciable events occur in these windows, there is no risk of washing out short-lived events. By comparison, the median filter is the first step in the filter pipeline proposed by [5], and consequently such features are likely to be missed.

In the ideal case, segments between proposed events are comprised of single states, and replacing the intermediate values with the mean is all that is necessary. It is very often the case, however, that additional considerations need to be made.

A. Edge cases

Figure 1 (left) shows a first example of an edge case, where the rise to steady state of a given appliance is slow enough relative to the sampling rate that the changes in demand are all sub-threshold. As a solution to this, a histogram of the intermediate samples is examined for well-defined peaks for each window. If there is a clear separation between power draw values in the window, a threshold is placed between them, and each 'sub-state' is individually replaced with the mean value.

Figure 1 (center) shows an additional case, where even with wavelet denoising, events are overestimated relative to ground truth due to large amounts of noise. This is often the case for appliances such as heat pumps, where y_{ss} and consequently the noise is large, and the approach to steady state is slow. A solution to these cases is to check that a linear fit over the window involves a correlation above some threshold, and at the same time, a slope above some threshold. This separates these cases from the previous ones, where often the correlation will be large, but the slope will not. If the segment meets these criteria, we note the time-step of this segment, and ask the same of the next window. We do this until the criteria are no longer met, and finally fill-backward the y_{ss} value at the end segment meeting these criteria. The result is a well-defined y_{ss} value for this particular heat-pump state.

Sharp transients are also undesirable for this type of representation, which vary widely in terms of width and height. An example is shown in figure 1 (right). To deal with these cases, we propose a post-processing step wherein the positive slope events whose next negative slope event within some sample threshold (say, 40 samples) has a large difference in value (i.e. an uncompensated positive slope event), are removed and the next y_{ss} value is filled backward. In other words, if the resulting squared signal falls within 40 s of rising and the rise is much larger than the fall, we consider this a transient and chop it.

Finally, even in relatively flat segments, it is often the case that events are over estimated due to random spikes in noise. As a result, the mean values between what should be a single segment can be different and below the threshold requested by the user. An additional post-processing step is proposed to deal with the following 8 cases: a positively or negatively sloped, sub-threshold event is followed by either a positive or negative sub-threshold event or a positive or negative event larger than the threshold. The solution is to either fill forward from the value prior to the sub-threshold event or fill backward from the value subsequent to the next event. Each of the above cases require a different method in order to recover the ideal case mentioned in the previous section.

IV. Experiments

A. Experimental Setup

Data was taken from House-1, Block-1 of the RAE dataset [8], which provides nine days of data sampled at 1 Hz. We chose to implement our method in Matlab, in part due to its wavelet denoising package, but also in anticipation of future work. All run-time tests under the heading 'Steady-state block filter' ran on a Mac Pro (Late 2011 model) with a 2 GHz Intel Core i7 processor and 4 GB of memory.

B. Experimental Results

In the interest of comparison with the filter pipeline proposed by [5], we used the same 5400 s sample consisting of the clothes dryer, the fridge, and the furnace. The raw aggregate and block-like representation are shown in figure 2. Table I shows the run-time comparison of the current method with that of the filter pipeline in [5]. Over 1000 trials, the present method performs at an average of over 6000 times faster (noting of course the differences in hardware). This shows the ability of our method to produce the desired result in an online setting. For completeness, the same knapsack algorithm developed by [5] was used on the block-like representation produced by the present method, and the results are tabulated in table II. As shown, the current method more accurately captures the energy of all tracked appliances, and outperforms on disaggregation accuracy scores across all appliances.

TABLE I
Run-time Comparison

Steady-state block filter		**Filter pipeline [5]**	
Process/Step	Time (sec)	Process/Step	Time (sec)
		Median Filter	1.6
Wavelet Denoising	0.0067	Bilateral Filter	12.7
Block generation	0.1267	Anisotropic Filter	0.1
Post-processing	0.0098	Edge-Preserving Filter	875.4
		Edge Sharpening	0.8
Total	0.1432	Total	890.6

To demonstrate the generality of this method to more complicated signals, figure 4 shows two examples of raw and block-like representations of mains data, illustrating some of the edge cases discussed previously.

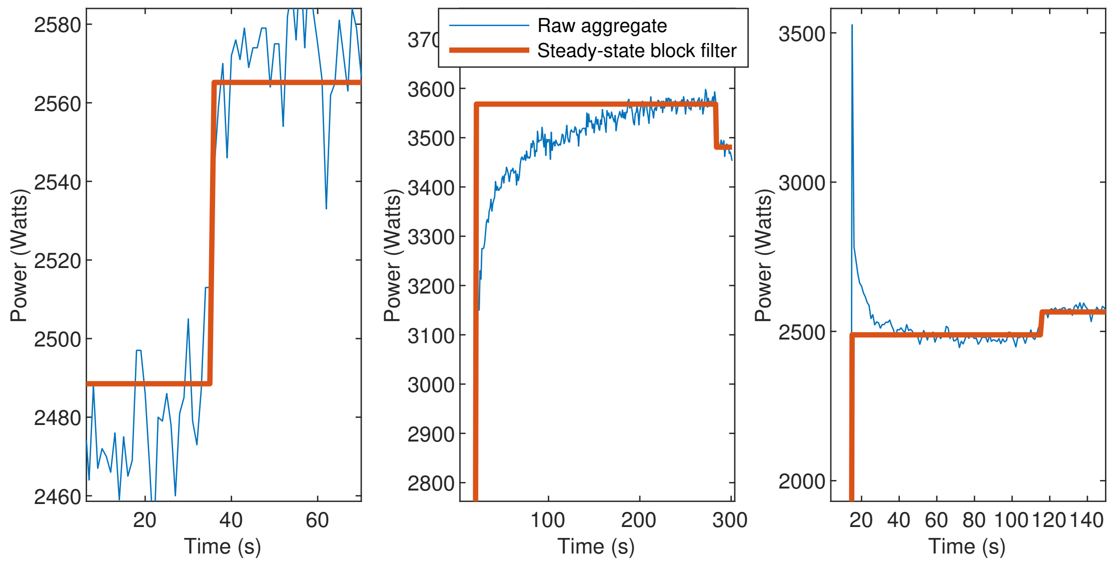

Fig. 1. Edge cases in order of presentation in text from left to right.

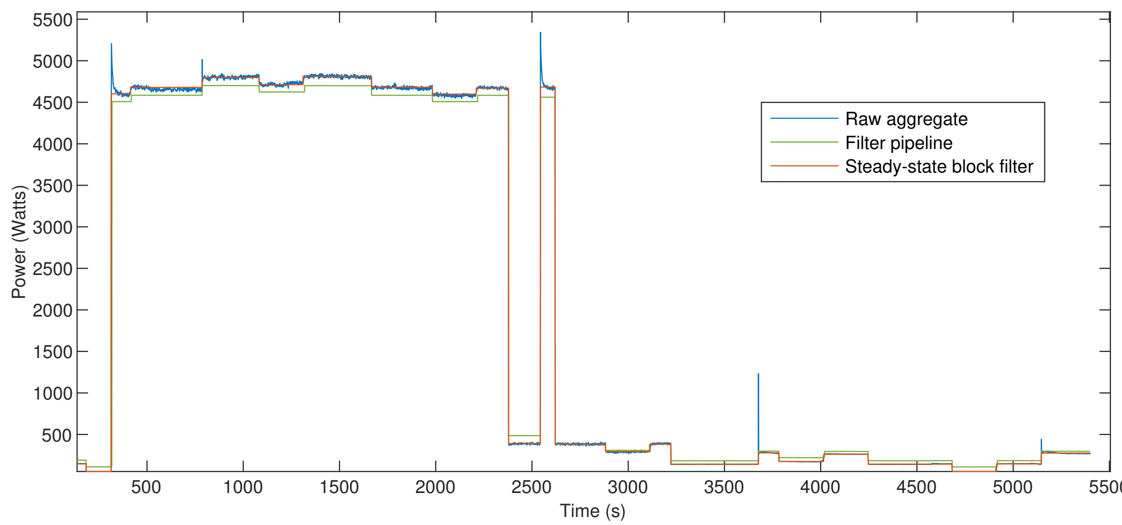

Fig. 2. Sample used in present work ('Steady-state block filter') and in [5] ('Filter pipeline').

V. Conclusions

We have demonstrated a significantly improved method for producing a clean, block-like representation of non-trivial raw data such that events larger than some specified threshold are separated by well-defined power draw values. Many current methods in NILM benefit from or require such a representation. Relative to a previously published method, our current method provides more reliable filtering at a fraction of the run-time, allowing its use in disaggregation methods requiring online use (assuming down-sampled reporting). Inevitably, more edge cases will become apparent as testing and iteration continues, but this method has merits of simplicity in both concept and implementation, and already provides consistent filtering of the dominant appliances contributing to overall energy consumption.

References

[1] G. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, Dec 1992.

[2] S. Makonin, "Investigating the switch continuity principle assumed in non-intrusive load monitoring (nilm)," in *2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, 2016, pp. 1–4.

[3] C. Klemenjak, A. Reinhardt, L. Pereira, S. Makonin, M. Berges, and W. Elmenreich, "Electricity consumption data sets: Pitfalls and opportunities," in *The 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, ser. BuildSys '19. ACM, 2019.

[4] D. Murray, L. Stankovic, V. Stankovic, S. Lulic, and S. Sladojevic, "Transferability of neural network approaches for low-rate energy disaggregation," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 8330–8334.

[5] A. Rodriguez-Silva and S. Makonin, "Universal Non-Intrusive Load Monitoring (UNILM) Using Filter Pipelines, Probabilistic Knapsack, and

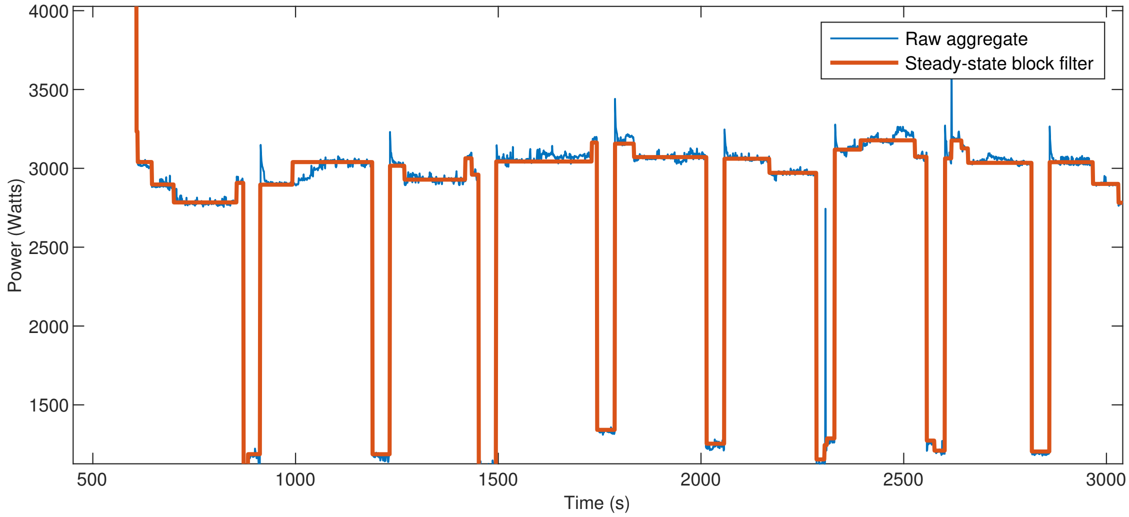

Fig. 3. Comparison of raw mains aggregate data and steady-state block filter output.

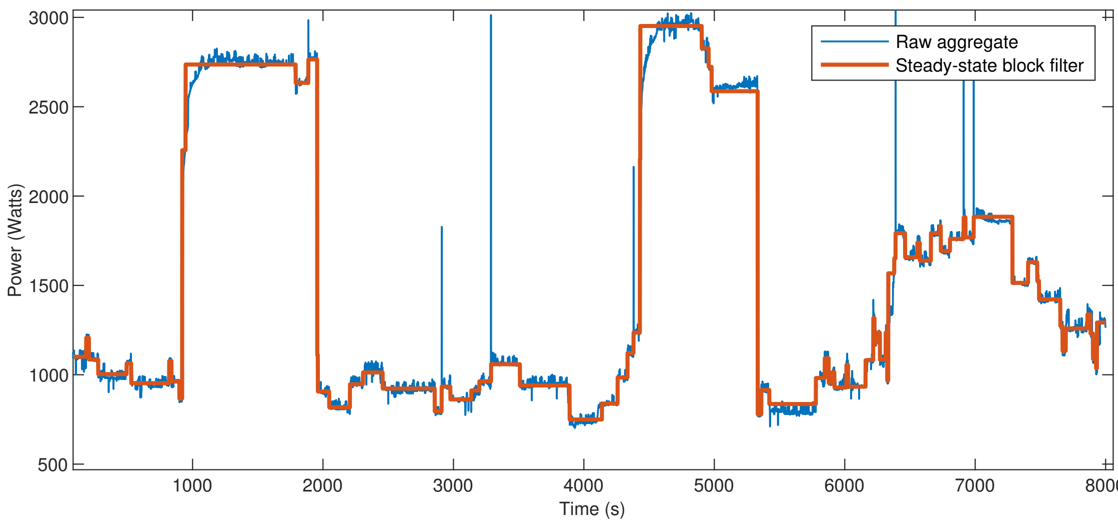

Fig. 4. Comparison of raw mains aggregate data and steady-state block filter output

TABLE II
COMPARISON OF ENERGY TRUTH/FILTERED/TRACKED (IN KWH)

Steady-state block filter				
Appliance	**G.Truth**	**Filtered**	**Est/Tracked**	**Truth *vs* Est**
Clothes Dryer	2.753	2.751	2.698	98.0%
Fridge	0.063	0.063	0.061	96.8%
Furnace	0.174	0.174	0.165	94.8%
Aggregate	2.990	2.988	2.924	97.8%
Filter pipeline [5]				
Appliance	**G.Truth**	**Filtered**	**Est/Tracked**	**Truth *vs* Est**
Clothes Dryer	2.753	2.729	2.604	94.5%
Fridge	0.063	0.065	0.055	87.3%
Furnace	0.174	0.167	0.144	82.8%
Aggregate	2.990	2.961	2.803	93.7%

Labelled Partition Maps," in *The 11th IEEE PES Asia-Pacific Power and Energy Engineering Conference 2019 (APPEEC)*, 2019.

[6] S. Makonin, F. Popowich, I. V. Bajić, B. Gill, and L. Bartram, "Exploiting hmm sparsity to perform online real-time nonintrusive load monitoring," *IEEE Transactions on Smart Grid*, vol. 7, no. 6, pp. 2575–2585, 2016.

[7] R. Dong, L. J. Ratliff, H. Ohlsson, and S. S. Sastry, "Energy disaggregation via adaptive filtering," in *2013 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, Oct. 2013.

[8] S. Makonin, Z. Wang, and C. Tumpach, "RAE: The Rainforest Automation energy dataset for smart grid meter data analysis," *data*, vol. 3, no. 1, p. 8, 2018.