IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 October 2022, accepted 3 November 2022, date of publication 9 November 2022, date of current version 14 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3220801

RESEARCH ARTICLE

Energy Disaggregation of Stochastic Power Behavior

DAISY H. GREEN[1], **(Member, IEEE),**
AARON W. LANGHAM[2], **(Graduate Student Member, IEEE),**
DEVIN W. QUINN[3], **THOMAS C. KRAUSE**[2], **(Graduate Student Member, IEEE),**
STEVEN R. SHAW[4], **(Senior Member, IEEE), AND STEVEN B. LEEB**[2], **(Fellow, IEEE)**
[1]Department of Architecture, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
[2]Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
[3]U.S. Coast Guard, Alameda, CA 94501, USA
[4]Department of Electrical and Computer Engineering, Montana State University, Bozeman, MT 59717, USA

Corresponding author: Daisy H. Green (dhgreen@mit.edu)

This work was supported in part by the Office of Naval Research Naval Enterprise Partnership Teaming with Universities for National Excellence (NEPTUNE) Program, and in part by The Grainger Foundation.

ABSTRACT Nonintrusive identification of the energy consumption of individual loads from an aggregate power stream typically relies on relatively well-defined transient signatures. However, some loads have non-constant power demand that varies with loading conditions. These loads, such as computer-controlled machine tools, remain stubbornly resistant to conventional nonintrusive electrical monitoring methods. The power behavior of these loads can be modelled with stochastic processes. This paper presents statistical feature extraction techniques for identification of this fluctuating power behavior. An energy estimation procedure is presented and evaluated for two case studies: load operation on a shipboard microgrid and laboratory machine shop equipment.

INDEX TERMS Nonintrusive load monitoring, power monitoring, energy management.

I. INTRODUCTION

Monitoring the electrical behavior of electromechanical loads is useful for energy management, condition-based maintenance, and detecting faulty equipment. A nonintrusive load monitor (NILM) is a convenient and inexpensive tool for power monitoring. All loads connected downstream of the utility point are monitored with a single set of current and voltage sensors [1], [2]. The advantage of nonintrusive load monitoring relies on the ability to disaggregate the energy consumption of individual loads from the aggregate power stream. Loads can generally be divided into those that have approximately discrete steady-state levels during operation and those with variable power demand [3]. The identification of loads with discrete steady-state levels and the tracking of resulting power changes has been well-documented in literature [4], [5], [6], [7], [8], [9], [10]. These loads are often identified with either event-based or optimization-based techniques. Event-based methods rely on detecting transient behavior, i.e., when the power usage of a system quickly changes while transitioning between discrete states, then applying a multi-class classifier. Many deep learning techniques have been applied for event-based load identification, such as convolutional neural networks (CNNs) [11], [12], [13], long short-term memory (LSTM) [14], and gated recurrent units (GRUs) [15]. Optimization methods, such as hidden Markov models (HMMs) [8], [9] and mixed-integer linear programming [5], [10], attempt to find the set of energized loads that best fit the aggregate measurement. By assuming a constant power level at each load state, the energy consumption of individual loads can be computed by tracking the operating duration. For optimization-based techniques, the power consumption of each load state is assumed to be known from prior knowledge, either using a representative steady-state value from sufficient historical data [5], [6], [7] or using the rated power value [16]. For event-based techniques, the steady state can be calculated using a short window after the detected load event [17].

Loads with variable power demand do not have unique power consumption levels or a fixed number of discrete states

The associate editor coordinating the review of this manuscript and approving it for publication was Akin Tascikaraoglu.

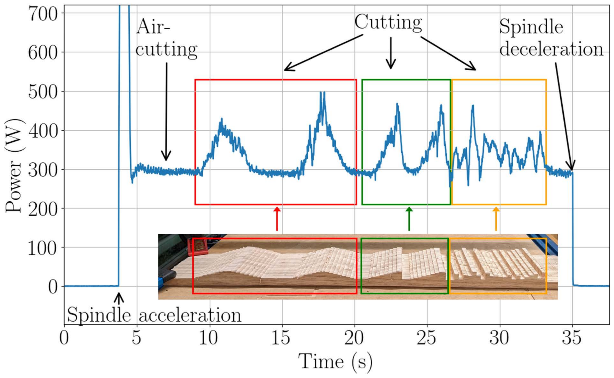


FIGURE 1. Power stream of CNC router showing power fluctuations corresponding to the cutting of areas highlighted in the wood board.

for the entirety of load operation. Accordingly, it is much more difficult to identify and track the operation of variable power loads with the same techniques used for loads that can be fully characterized with distinct, relatively repeatable power consumption transients. Devices can exhibit variable power demand in response to dynamic loading conditions. For example, machines like routers and mills have changing power demand based on cutting conditions, such as workpiece material and depth of cut [18]. Fig. 1 shows the real power stream as a CNC router makes a cut through the shown wooden board. The vertical position of the router cutting tool is held constant. The contour of the workpiece changes the axial depth of cut. When the cutting tool is freely rotating and not engaged in the workpiece, there is an approximately constant power consumption level. However, when the cutting tool engages the wood, the power varies significantly, and clearly resembles the workpiece contour. Power fluctuations due to changing load demand occur in response to understandable physical behavior; however, these fluctuations appear as unpredictable events in the power stream. The power behavior of these types of loads can be modelled with stochastic processes. Fluctuating power demands, e.g., in a CNC machine tool, can also correspond to loaded conditions and mechanical processes that introduce wear into the system. Thus, tracking the duration of loaded operation can be useful for load condition monitoring and use tracking.

Advances in pattern classification and deep learning have enabled automated feature extraction and classification on high-dimensional data. On the surface, identification of stochastic power behavior would appear to be an ideal problem domain for such deep classifiers. Classifiers such as CNNs can extract features and "learn" the stochastic behavior of the loads of interest. However, these approaches require a relatively large amount of training data and effort to obtain high generalization [19]. A practical NILM in industrial sites will likely need to collect its own training data, making training data a scarce resource. Rather than use high-dimensional time-domain data for classification, the proposed system extracts lower-dimensional features that describe the observed distributions of the power values. This makes it possible to avoid the burdens of requiring large training datasets and long training times associated with deep learning classifiers.

This paper presents a method for the disaggregation of fluctuating power behavior from more well-defined transients in an aggregate power stream. The fluctuations typically have certain statistical properties characteristic of load operation. The presented method is based on statistical features. The method is specifically suited for loads which exhibit variable power consumption due to changing load demand. Identification of statistical events are incorporated in a new procedure for estimating individual load energy consumption. Laboratory and field results are presented from two case studies: load operation on a shipboard microgrid and laboratory machine shop equipment.

II. RELATED WORK

Variations in power draw can occur at multiple time-scales as a result of different mechanical processes [20], [21], [22]. As presented in [20], power consumption events can be distinguished into geometric, statistical, and continuous events. Discrete transitions between states of a load are referred to as geometric events, since the salient features are extracted from the shape of the transient. There is always a steady-state change after a geometric event. Typically, on/off loads and multi-state loads consist of geometric events. Statistical and continuous events both have variable power demand. The fluctuating power variations that comprise statistical events represent periods of variable loading. For example, in Fig. 1, the spindle acceleration and deceleration are geometric events, air-cutting is a discrete power consumption state, and the cutting operations are comprised of statistical events. Statistical events do not necessarily indicate that a load has transitioned to a different steady-state level. The identification of statistical events enables the disaggregation of power during the event itself and the detection of any change in steady state as a result of the event. A continuous load can consume any amount of power in a given range. For example, power electronic loads such as variable speed drives (VSDs) and light dimmers may exhibit smoothly variable power demand. That is, a continuously variable load does not have any set steady-state power consumption level. The power variations of interest in this work (statistical events) occur on slower time-scales than the main load on- and off-events (geometric events), but at faster time-scales than the smoothly varying power demand of continuous events.

A. MULTI-STATE LOADS

Multi-state modeling techniques have been widely used to address appliances that have various operational states [5], [6], [7], [8], [9], [23]. Many multi-state modeling methods assume a constant, or discrete power at a given state. That is, these methods do not consider variations in power demand during a load state. To address this, the work in [23] presents an energy disaggregation technique that estimates the power of devices at every sampling instant by formulating the problem as a constrained optimization problem. The method first determines device states using a hidden Markov model (HMM). Then, each load's power is estimated by maximizing the total probability of the observations while constraining their sums to equal to the total measured aggregate power. This method assumes that the observations of a given load state converge to a Gaussian distribution. The authors in [23] state that this assumption, although reasonable for many loads, does not hold for loads with continuously variable power. As will be demonstrated in this paper, the assumption also does not hold for loads with stochastic power behavior (i.e., statistical events). The method in [23] also assumes that the distributions are stationary and well characterized from previously recorded observation data. However, in many practical scenarios, a load's power characteristics will drift over time [24].

B. CONTINUOUS LOADS

References [3] and [25] present methods for identifying the smoothly varying power of power electronic loads (i.e., continuously variable loads). These methods rely on significant higher-order current harmonics to distinguish the fundamental-component power consumption of power electronic loads from that of loads with constant steady-state power demand. However, these higher harmonic signatures are not always present for all loads. Furthermore, these methods assume that only a single power electronic load is present in the aggregate stream. If there are multiple power electronic loads energized, only the summed fundamental power demand of all the power electronic loads can be disaggregated from that of loads with constant steady-state power demand. Reference [26] presents an optimization-based method for identifying the power consumption of power electronic loads. The method relies on having an accurate characterization of the magnitude and phase of the fundamental current component of individual loads for all operating ranges. The disaggregation of continuously varying power is out of the scope of our proposed work. However, our proposed work still applies to a power electronic load if it is kept at a single speed setting, but exhibits variable power demand due to dynamic loading conditions. Higher-order current harmonics can serve as additional features in this scenario.

C. STATISTICAL EVENTS

A new load identification method for stochastic power behavior was presented in [20] based on statistical features. However, this method did not consider that the fluctuation variance can change based on factors such as the steady-state power consumption of the load or the type of material being cut by a CNC machine. That is, the magnitude of fluctuating power may not be a reliable feature. This paper addresses this issue by presenting a method for evaluating stochastic behavior independent of load and operating condition. Furthermore, the work in [20] does not incorporate the identification of stochastic power behavior into any energy estimation methods. This paper expands on [20] by using the identification of stochastic behavior for accurate energy estimation and condition tracking.

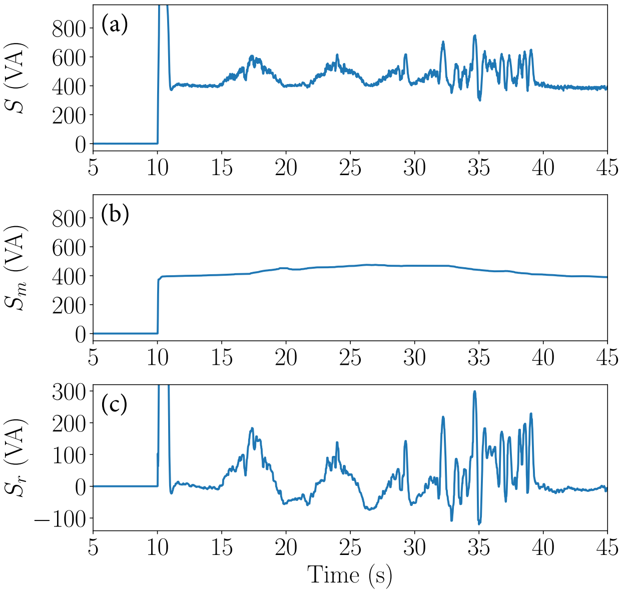


FIGURE 2. CNC router: (a) apparent power stream, (b) median stream, (c) residual stream. Note that the y-axis scale in (c) is different from (a) and (b).

III. METHODOLOGY

This section presents a method for the disaggregation of statistical events of multiple loads from an aggregate power stream. Just as the identification of geometric events relies on an event detector and a feature extractor, there is an analogous event detector and feature extractor for statistical events, as described in this section. First, the event detector is used to find events. Then, at each event, a set of features is extracted and used for load classification. This section also presents an energy estimation procedure using the classified statistical events.

A. STATISTICAL EVENT DETECTION

For illustrative purposes, consider the fundamental real power (P) and reactive power (Q) streams, which in this work are sampled at a frequency of 60 Hz. From P and Q, the apparent power (S) can be calculated as $S = \sqrt{P^2 + Q^2}$. The apparent power stream is used in this work for event detection. First, a seven-point rolling median filter is run on the S stream to reduce noise. Then a longer rolling median filter, typically on the order of seconds or minutes, is run. The resulting stream is the median stream, S_m. Then, S_m is subtracted from the original S stream, resulting in the residual stream, S_r. That is, the residual stream is calculated as, $S_r = S - S_m$. An example apparent power stream, median stream and residual stream are shown in Fig. 2 for a CNC router, in which the median filter length is 20 seconds. The median filter length should be set by the user based on the expected time-scale of power fluctuations. The median filter

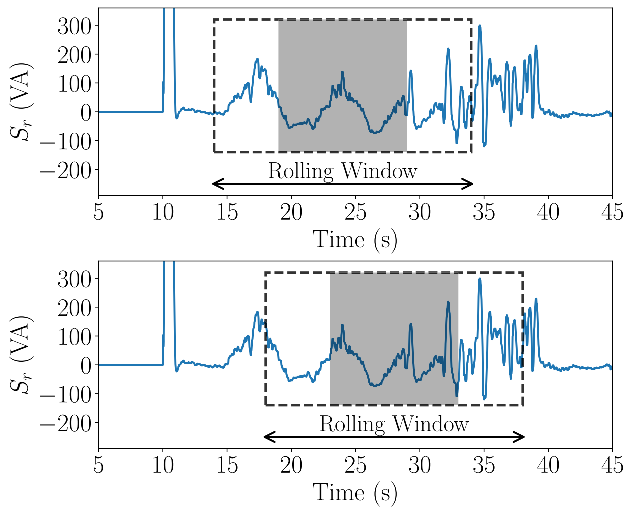


FIGURE 3. Rolling window of CNC router residual stream for statistical event detection. The shaded regions represent the middle 50% of each window.

should preserve sharp edges due to geometric events, but remove the statistical event power fluctuations [20]. As a result, the residual stream should contain the statistical events of interest.

Residual streams are extracted from a rolling window of input data. The rolling window length and overlap are parameters that the user can set. For instance, Fig. 3 shows a CNC router S_r stream for two instances of a 20-second rolling window with 80% overlap. The standard deviation, denoted here as σ, is calculated for the middle 50% of each S_r stream window, as indicated by the shaded regions in Fig. 3. Using the middle 50% of the window decreases the likelihood that large fluctuations occur only on the edges of the window. If σ is greater than a user-defined threshold, it indicates significant activity of interest (i.e., a statistical event is detected). For every detected statistical event, further statistical feature extraction is performed, as will be described in Section III-B. The value of σ should be chosen to be larger than the standard deviation of the aggregate power stream measurements when loads are in steady state. The rolling window length should be selected to be long enough to capture the statistical characteristics of the stochastic power behavior. At the same time, it should be short enough so that the stochastic power behavior can be expected to occur for the entire window duration. That is, for a region of stochastic power behavior, it is desirable to have multiple rolling windows containing the statistical behavior in the region. For instance, if the desired number of rolling windows for each statistical region (R), the overlap percentage of each window (v), and the expected duration of the statistical region ($T_{statistical}$) are known, the rolling window duration ($T_{rolling}$) can be computed as follows:

$$T_{rolling} = \frac{T_{statistical}}{R(1-v)+v}. \tag{1}$$

Since the rolling window length and median filter length are both set based on the expected time-scale of power fluctuations, they should be similar in length. In the case studies in Section IV, the rolling window length was set equal to the median filter length.

B. STATISTICAL FEATURE EXTRACTION AND CLASSIFICATION

To provide greater load separation in the feature space, the feature vector for classification uses features extracted from the real and reactive power streams. The residual streams for real and reactive power, P_r and Q_r, respectively, are calculated using the same process as described for the apparent power stream in Section III-A. This work uses four streams for feature extraction: real power residual stream (P_r), reactive power residual stream (Q_r), first-order difference of the real power residual stream (P'_r) and first-order difference of the reactive power residual stream (Q'_r). These features relate to the physics of load behavior. Other power streams could be relevant in some scenarios, such as the higher-order harmonic current spectral envelope streams when multiple power-electronic loads are monitored. As mentioned, the magnitude of power fluctuations may depend on load and operational condition. However, if the data is normalized to a constant range, the stochastic behavior can be evaluated independent of the change in overall fluctuation magnitudes. Min-max normalization is performed on these four data streams individually for each window. The range of the data is transformed into [0, 1] for each stream window x through the transformation,

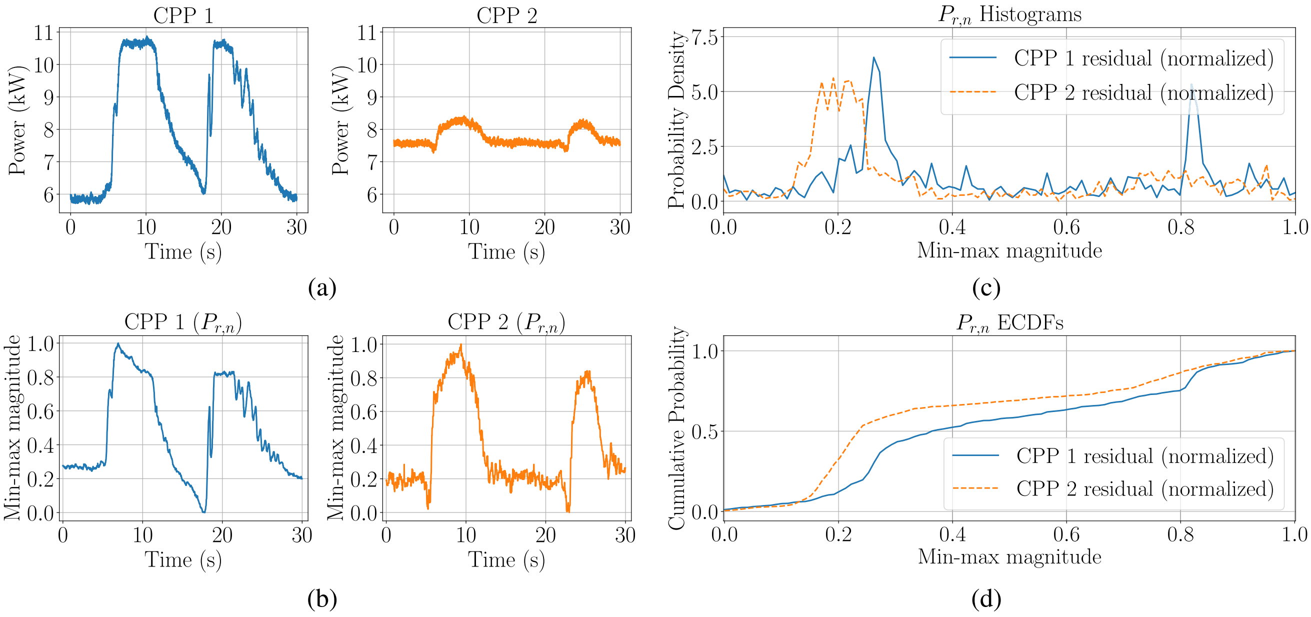
$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}. \tag{2}$$

Fig. 4a shows two 30-second windows of the original P stream of a shipboard controllable pitch propeller (CPP) pump. Fig. 4b shows the resulting normalized residual streams ($P_{r,n}$) using a 30-second median filter. For each window, an empirical cumulative distribution function (ECDF) is computed on the normalized stream, x_{scaled}, by creating a histogram of the data values, and then applying a cumulative sum. Just as a histogram is the empirical estimate of a probability density function (PDF), an ECDF is the empirical estimate of a cumulative distribution function (CDF). A histogram is created by grouping the data into bins. Each bin is a unique interval such that the union of the bins covers the range [0,1] and all bins have the same width. The number of bins should be chosen by the user. The value of the ECDF for a given bin b, denoted as $\hat{F}_b$, is given as the relative frequency of all observed values of x_{scaled} being less than or equal to the value of x_{scaled} represented by bin b (denoted as x_b in Eq. (3)). This is given by the following for each data window of length N:

$$\hat{F}_b = \frac{1}{N}\sum_{i=1}^{N} I(x_i \leq x_b) \tag{3}$$

where I is the indicator function, given by

$$I(x_i \leq x_b) = \begin{cases} 1, & \text{if } x_i \leq x_b \\ 0, & \text{if } x_i > x_b \end{cases}. \tag{4}$$



(a)

(c)

(b)

(d)

FIGURE 4. **Process for calculating ECDF curves for two example windows. (a) Real power streams. (b) Normalized real power residual streams ($P_{r,n}$). (c) Histograms of $P_{r,n}$ streams. (d) ECDFs of $P_{r,n}$ streams.**

Fig. 4c and Fig. 4d show the histograms with 100 bins and resulting ECDF curves, respectively, for the two windows in Fig. 4b. The two windows have different steady-state load levels. Due to the mechanical process causing the fluctuations in power for this load, generally as the steady state increases, the magnitude of the fluctuations in power decreases. Even though the overall magnitudes of the fluctuations are drastically different, the ECDF curves are similar using the normalized data.

The ECDF is used as a feature vector representation of the statistical properties of the window of interest. Computing the Euclidean distance (l_2 norm) between two ECDFs is equivalent to the Cramer-von Mises test, a useful metric for estimating distribution equality [27]. An alternate choice of vector norm could be used if desired, e.g. the l_∞ norm, which would correspond to the Kolmogorov-Smirnov test [27]. This work uses the k-nearest neighbors (k-NN) classifier, since many distance metrics are applicable. The feature vector used as input to the k-NN classifier in this work is a concatenation of four ECDF curves, computed for $P_{r,n}$, $Q_{r,n}$, $P'_{r,n}$, and $Q'_{r,n}$. The classification process involves finding the k nearest training data points to the input feature vector, using the Euclidean distance. The input is classified to the class whose points makes up the plurality of these k nearest neighbors. Typically, choosing a larger value of k makes the classifier less likely to overfit, but ignores more local patterns in the data [28]. A larger value of k will also require more computation in searching for the neighbors of an input point. For binary classification (i.e., when there are only two classes), choosing k to be odd results in there never being a tie between classes. Neighbors can also be weighted based on the distance to the input feature.

C. ENERGY ESTIMATION

Large fluctuations in power draw need to be attributed to the correct load for accurate energy estimates. These large fluctuations are not accounted for when using event-based algorithms that assume approximately discrete steady-state levels [17], [29]. Using the proposed statistical features, once the stochastic behavior has been classified, the original power stream in the windows of interest can be numerically integrated to estimate the energy of each load. In addition, loads can exhibit stochastic behavior in the transition between steady states. By identifying the difference in steady-state power consumption for the load exhibiting stochastic behavior, its steady-state power can be updated in real time. As an example, consider the shipboard bilge and ballast pump shown in Fig. 5, with identified stochastic behavior labeled. The steady-state real power levels before and after the stochastic behavior are wildly different. This new steady-state value can be tracked and assigned to this load so that an incorrect steady-state value is not used in energy estimation.

The proposed algorithm to process the power stream into windowed events is given in Algorithm 1. This algorithm calls the function described in Algorithm 2 in order to disaggregate the energy for each load in the stream. A rolling window of user-defined length (window_length in Algorithm 1) is run through the power stream, and geometric and statistical event detectors are run. Note that window_length is equivalent to N from Eq. (3). The geometric event detector is responsible for identifying load turn-on and turn-off events, as well as transitions between discrete states. Here, it is assumed a geometric event classifier has already been trained. Although not in the scope of this paper, many geometric event detectors

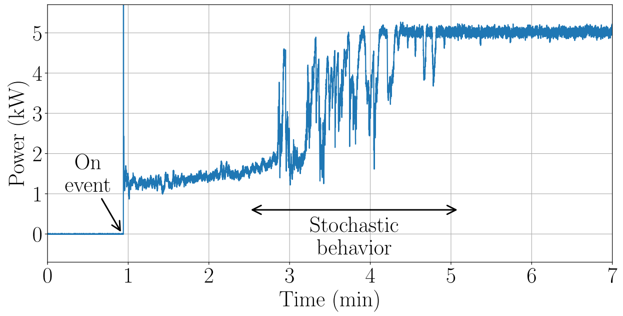


FIGURE 5. Power stream of a shipboard bilge and ballast pump with stochastic behavior in the transition between steady states.

Algorithm 1 Algorithm for Processing Power Stream Into Windowed Events

```
Input: power_stream
Input: load_list
Input: window_length
Input: geometric_length
Input: statistical_increment
 1: $n \leftarrow$ window_length
 2: while $n \leq$ length(power_stream) do
 3:     window $\leftarrow$ power_stream[$n$ - window_length:$n$]
 4:     if CheckForGeometricEvent(window) then
 5:         event_load $\leftarrow$ GeometricClassifier(window)
 6:         Update event_load.ss
 7:         $j \leftarrow$ geometric_length
 8:     else if CheckForStatisticalEvent(window) then
 9:         event_load $\leftarrow$ StatisticalClassifier(window)
10:         Update event_load.ss
11:         $j \leftarrow$ statistical_increment $\cdot$ window_length
12:     end if
13:     $j \leftarrow 1$
14:     EnergyEstimation(event_load, $i$)    ▷ Run Algorithm 2
15:     $n \mathrel{+}= j$
16: end while
```

Algorithm 2 *EnergyEstimation()* Algorithm

```
Input: event_load (can be None)
Input: $j$, amount to increment window
if event_load is None then
    for load in loads do
        if only load is operating then
            load.energy $\mathrel{+}=$ Trap(power_stream, $j$)
        else
            load.energy $\mathrel{+}=$ Rect(load.ss, $j$)
        end if
    end for
else
    for load in loads except event_load do
        load.energy $\mathrel{+}=$ Rect(load.ss, $j$)
        event_load.energy $\mathrel{-}=$ Rect(event_load.ss, $j$)
    end for
    event_load.energy $\mathrel{+}=$ Trap(power_stream, $j$)
end if
```

are applicable [20]. The statistical event detector, described in Section III-A, is responsible for identifying regions with large load power fluctuations. The time index of the end of the rolling window is designated as n in Algorithm 1 and is incremented by a number of time indices as determined by the output of the geometric and statistical event detectors. For the first window, n is the same as the window length, as designated in Line 1 of Algorithm 1. If a geometric event is detected in the window, the rolling window (and n) is incremented by the length of the geometric window (geometric_length in Algorithm 1). If a statistical event is detected in the window, the rolling window is incremented by statistical_increment times window_length. Here, statistical_increment is one minus the amount of statistical window overlap. If no events are detected in the window, the window is incremented by a single time index. Line 15 in Algorithm 1 shows the incrementation of the time index n.

The algorithm keeps track of the current steady-state real power of each load (denoted as load.ss in Algorithms 1 and 2). This value is updated for a given load whenever a geometric or statistical event from that load is detected. For geometric events, the steady state after an on-event is calculated as the change in mean value of a user-defined length window (e.g., 0.2 seconds) at the end and start of the geometric window. Turn-off events detected by the geometric event detector are assumed to change the load's steady-state power to zero, rather than using the difference computed across the window. An assumption is made that the stochastic behavior occurs in superposition with the steady-state power consumption of all other energized loads and that the steady-state power of the other loads has remained constant. That is, the steady state after a statistical event is the aggregate power at the end of the window (i.e., the mean of last 0.2 seconds of the window) minus the stored steady-state values of the other energized loads. The statistical event detector will only classify and update the steady state if the classified load does not currently have a steady-state power of zero.

The energy estimation algorithm uses a combination of rectangular and trapezoidal integration [30], denoted in Algorithms 1 and 2 as *Rect*() and *Trap*(), respectively. In this work, only real power is integrated, so that the result has physical meaning as useful work done by the system. However, reactive and apparent power can also be integrated with the same process. When there is only one load operating, trapezoidal integration is performed, and the resulting energy is added to the operating load's total energy. When a statistical window is identified as a specific load, all other loads' energies are calculated with rectangular integration of their steady-state real power (i.e., load.ss). Trapezoidal integration is performed and the result is added to the load's total energy. The sum of the energy calculated for the other loads with rectangular integration is subtracted from the identified load's energy total, to remove the contribution of the other loads

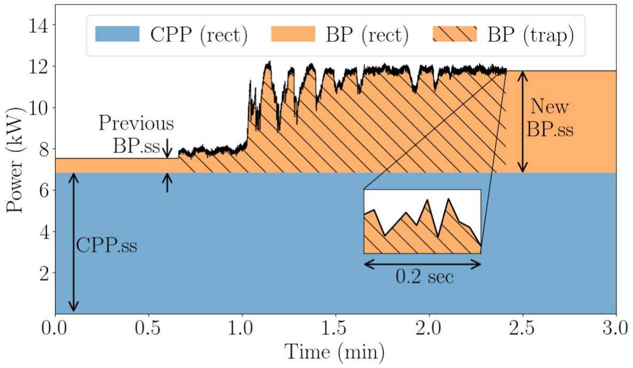


FIGURE 6. Demonstration of the energy estimation algorithm tracking a load's new steady state after identifying statistical event regions.

added in the trapezoidal step. In all other cases, rectangular integration uses each load's stored steady-state real power. An instantaneous estimate of disaggregated power can be computed similarly. For each window, each load's power is assumed to be its current steady-state power, unless it was responsible for a geometric or statistical event in the window. In this case, its power over the window is assumed to be the power stream over the window minus the sum of the current steady-state powers of the other loads. The total duration of statistical event operation for each load can be tracked using the identified statistical event windows.

An example of this process is illustrated in Fig. 6, in which there are two loads operating, a controllable pitch propeller (CPP) pump and a bilge and ballast pump (BP). In the figure, the CPP is the base load with a steady state indicated as "CPP.ss." Since there are multiple loads energized and no CPP geometric or statistical events in this window, the CPP steady state is assumed constant and its energy is calculated using rectangular integration. For the BP, the area indicted by the shaded region has been identified as statistical event windows. The steady state of the BP prior to the statistical event region is indicated by "Previous BP.ss." The steady state of the BP after the statistical event region is indicated by "New BP.ss" and was calculated by taking the mean of the last 0.2 seconds of the statistical region, as highlighted with the zoomed-in inset, and subtracting CPP.ss. Energy estimation of the BP statistical region is calculated using trapezoidal integration.

The most computationally challenging aspect of the proposed method is the long median filter for calculating the residual stream. For instance, for a 30-second median filter of three streams (P, Q, and S) sampled at 60 Hz, 5400 samples must be stored at a given time. This will also introduce a delay of half the length of the median filter to the output residual streams and thus to the entire energy estimation process. However, we claim that this delay is not prohibitive for almost real-time energy usage information. Our previous work has shown that a NILM with modern computing power is capable of running such a filtering system in real time [20].

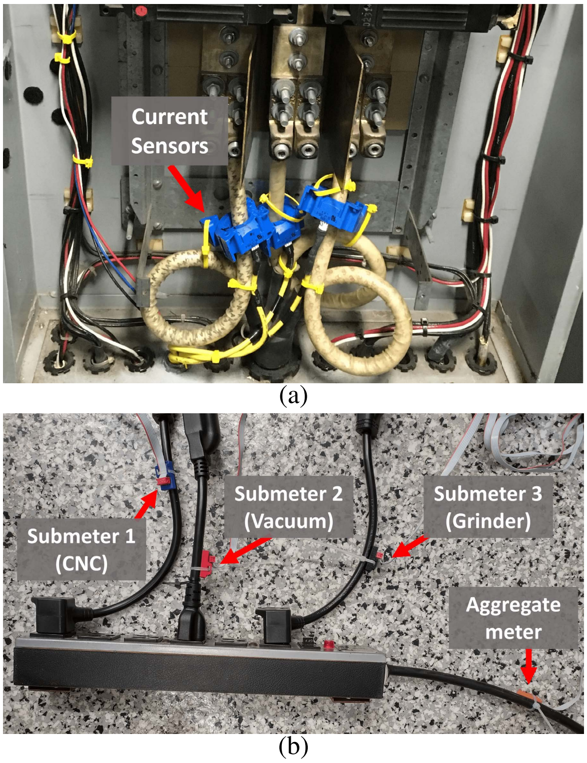


FIGURE 7. Experimental setups: (a) USCGC SPENCER electrical panel interior with installed current sensors. (b) Non-contact meters in laboratory machine shop.

IV. EVALUATION AND RESULTS

To evaluate the proposed statistical event detection and energy estimation methods, results from two case studies are presented in this section: shipboard loads and machining equipment. In both cases, rolling statistical windows were set with 80% overlap.

A. EXPERIMENTAL SETUP

The presented case studies use two different data acquisition setups: "contact" and "non-contact" NILMs. The first case study uses a contact NILM installed on the starboard electrical subpanel of the engine room of United States Coast Guard Cutter (USCGC) SPENCER. This panel supplies power to auxiliary equipment necessary for maintaining operational readiness of the ship's starboard main diesel engine and ship-service diesel generator, as well as several other loads critical for ship operation [20]. This contact NILM setup uses the LEM LF-305 transducer for current measurement, as shown in Fig. 7a, and wires into a spare breaker for voltage measurement. This setup does not allow for submetering of individual loads, so load events, both geometric and statistical, were hand-labeled by a domain expert. For the second case study, both a contact NILM and non-contact NILMs were used in a laboratory machine shop. First, a contact NILM was used and loads were individually energized for use as training data. This contact NILM setup used the LEM LA-55 transducer for current measurement. Then, to simultaneously monitor the aggregate power stream and to

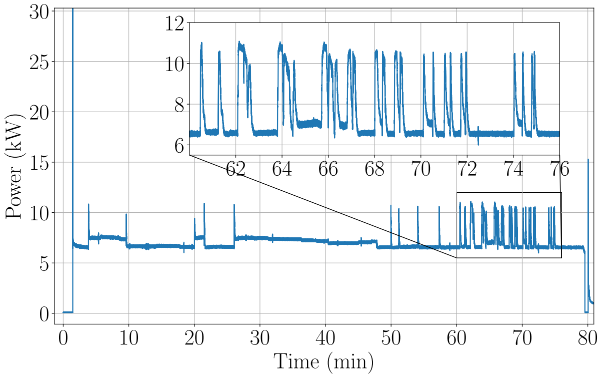


FIGURE 8. Power draw of an example run of the CPP pump with a zoom-in on power "surging" [21], [22].

individually submeter each piece of equipment, non-contact meters (as described in [31]) were installed. The sensing device for a non-contact NILM can be placed around the electric cable, permitting current and voltage sensing without service interruption. This enables ground-truth verification while multiple loads are energized. Fig. 7b shows the non-contact NILM setup in the laboratory machine shop. The contact and non-contact NILMs sample current and voltage streams at 8 kHz and 3 kHz, respectively. The current and voltage streams are processed into real power (P), reactive power (Q), and higher-order current harmonics at sampling rate equal to utility frequency of 60 Hz [32].

B. SHIPBOARD LOADS

On the monitored shipboard subpanel, there are two loads that have stochastic behavior: the bilge and ballast pump (BP) and the controllable pitch propeller (CPP) pump. The BP, which is rated at 5.6 kW, is used for emptying machinery space bilges of excess water in an emergency and for taking on ballast water for stability purposes [33]. When pumping bilges and ballast tanks, operators try to get the tanks and bilges to the lowest level possible, and as a result, the pump takes in a mixture of air and water. After the pump is turned off and suction is shifted to a new tank, the air remains in the system, resulting in a prolonged start sequence in which the pump draws a variable amount of power. It has been observed that the initial steady state can be as small as one-fifth of the expected steady-state level. As shown in Fig. 5, the pump typically reaches the expected steady state, but it may take time on the order of minutes. This large discrepancy complicates energy disaggregation using purely geometric methods. The monitored CPP pump is rated at 7.5 kW. The pump is part of the CPP system, which is operated underway to provide the vessel greater maneuverability [33]. The monitored CPP 'C' pump is an electric hydraulic pump that supplements a separate gear driven pump in order to provide pressurized hydraulic oil to the CPP system and maintain hydraulic control pressure at the propeller. Hydraulic control valves maintain system operating pressure based on demand. Fig. 8 shows the real power of one example run of a monitored CPP pump. There are "surges" in power, as highlighted in the zoomed-in window. These surges are a result of the CPP pump compensating for the extra pressure required during ship maneuvering. Comparing the windows in Fig. 4, if the system is already at the required pressure and the CPP pump is at the required power, the CPP pump does not need to surge as much to compensate.

For these loads, the median filter length and rolling window length were both set to 30 seconds. This window length and median filter length were chosen to generally match the time-scale of observed power fluctuations. For training, instances of individual operation of the CPP pump and BP were used. Rolling windows were used with a standard deviation of $\sigma = 100$ W as the threshold for identifying windows with statistical activity. This threshold for σ was chosen to be larger than the standard deviation in the aggregate power stream when loads are at steady state. In total, the dataset has 612 windows of the CPP and 788 windows of the BP. The data was randomly split into 60% training data and 40% testing data with data stratification to allocate samples evenly based on sample class. A k-NN classifier was trained with $k = 3$ without weighting. The dataset split and training was run 10 times for verification, with the results for the testing sets averaged and shown in Table 1. The results are presented as the average F_1, precision, and recall scores, with σ_{F_1}, σ_{Pr}, and σ_{Re} showing the standard deviation of the runs. Precision, recall, and F_1 scores of 1 indicate perfect performance in identifying a specific class. The high average scores and small standard deviations indicate that the proposed method has good consistency between runs.

For comparison, a deep neural network (DNN) was also trained with the same windows of power data, using the raw 30-second P and Q signals as the input features (with the mean value of each window subtracted). This DNN used four layers with 500, 200, 100, and 1 neurons each, and was trained using Adam-optimized backpropagation with binary crossentropy as the loss function. The results averaged across 10 runs are shown in Table 1. The low average scores and large standard deviations indicate inconsistency between runs. The poor performance is likely due to the models overfitting to the training data because of the high input dimensionality and relatively small training dataset.

The uniqueness of stochastic behavior can be explained by the physical mechanisms. For instance, Fig. 9 shows the average ECDF curves for the CPP and BP $P'_{r,n}$ streams. The lines represent the average of all ECDFs for each class in the dataset and the shaded regions represent one standard deviation of the ECDF values. The figure shows that the mean value of the BP ECDF (represented by the orange dashed line) corresponds to larger min-max magnitudes than the mean value of the CPP ECDF (the blue solid line). Fig. 10 shows example time-domain windows for both the CPP and BP. Fig. 10a shows the normalized residual stream ($P_{r,n}$) and Fig. 10b shows the normalized first-order difference of the residual stream ($P'_{r,n}$). The CPP surge events have a sudden large increase in power but a slower decrease back to

TABLE 1. Shipboard loads statistical event classification.

Equipment	# of Windows	DNN $\mathbf{F_1}$ ($\sigma_{\mathbf{F1}}$)	DNN **Pr** ($\sigma_{\mathbf{Pr}}$)	DNN **Re** ($\sigma_{\mathbf{Re}}$)	Proposed (ECDFs + k-NN) $\mathbf{F_1}$ ($\sigma_{\mathbf{F1}}$)	Proposed (ECDFs + k-NN) **Pr** ($\sigma_{\mathbf{Pr}}$)	Proposed (ECDFs + k-NN) **Re** ($\sigma_{\mathbf{Re}}$)
CPP	245	0.54 (0.308)	0.56 (0.227)	0.64 (0.390)	0.97 (0.006)	0.97 (0.010)	0.98 (0.009)
BP	315	0.66 (0.199)	0.79 (0.144)	0.68 (0.303)	0.98 (0.005)	0.98 (0.007)	0.98 (0.008)

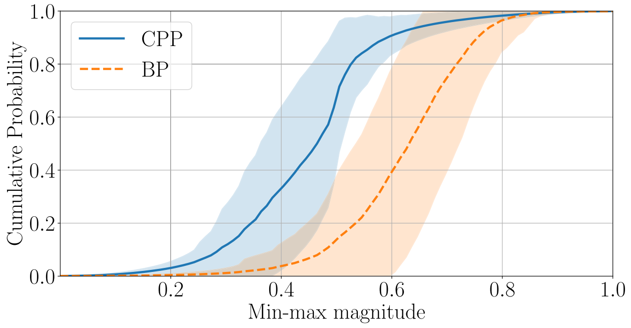


FIGURE 9. ECDF curves of the normalized CPP and BP first-order difference streams. The lines represent the average of all statistical windows in the dataset and shaded regions represent one standard deviation.

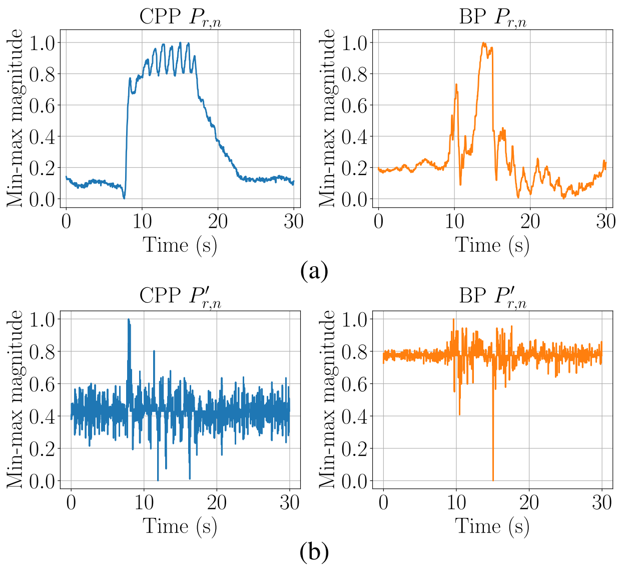


FIGURE 10. (a) Normalized residual streams. (b) Normalized first-order difference of residual streams.

steady state. That is, most of the large first difference values are positive. Due to these large positive "spikes," the average after min-max normalization will be less than 0.5. In contrast, the rapid fluctuations of the BP generally results in larger "spikes" in the negative direction, indicating many large magnitude negative first difference values. The average after min-max normalization is generally greater than 0.5.

The statistical classifier and energy estimation algorithm were run for a two-hour window of the starboard subpanel aggregate NILM stream in which the CPP and BP were both energized. The threshold for identifying statistical events was set to $\sigma = 100$ W. The aggregate power stream is shown in Fig. 11a. The estimated power of the CPP and BP using the proposed method is shown in Fig. 11b. The estimated power was also computed using the method in [23] and using rectangular integration. All the methods used the same geometric classifier. That is, it is assumed that all discrete state transitions have been accurately detected. The method from [23] (also described in Section II) estimates each load's power by maximizing the total probability of the observations while constraining the total power. This method relies on knowing the mean and variance of the power measurements for each load state. This assumes having a well-characterized load, typically using sufficient historical data. Two different state determinations were tested, where both the CPP and BP have: 1) on and off states, and 2) on, loaded, and off states. Note that the loaded "state" here corresponds to statistical event periods, as identified by the proposed statistical event classifier. The mean and variance for the on state were calculated from the steady-state behavior of load observations, and for the loaded state they were calculated from fluctuating power observations. For the on state, the calculated mean values for the CPP and BP were 6.7 kW and

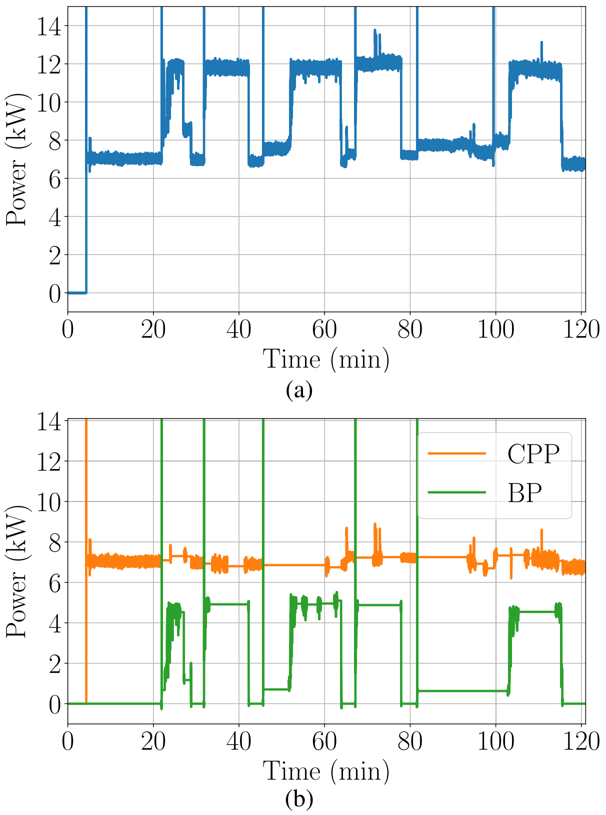


FIGURE 11. (a) Measured aggregate power. (b) Estimated individual load power.

TABLE 2. **Shipboard loads energy estimation.**

	CPP	BP	Total
Proposed	13.761 kWh	4.238 kWh	17.999 kWh
Method from [23] (2-state)	12.015 kWh	5.951 kWh	17.967 kWh
Method from [23] (3-state)	12.005 kWh	5.961 kWh	17.967 kWh
Rectangular	13.964 kWh	1.723 kWh	15.687 kWh

5.0 kW, respectively, with standard deviations of 0.09 kW and 0.07 kW. For the loaded state, the mean values for the CPP and BP were 7.1 kW and 3.8 kW, with standard deviations of 0.7 kW and 1.2 kW. As described in Section II, there are various ways to calculate the steady-state value for rectangular integration [5], [6], [7], [16], [17]. For comparison here, the steady-state value used is the one calculated after the load on-event. Table 2 shows the resulting energy estimates for the entire two-hour window for the individual loads and the total energy. This monitoring setup does not have individually submetered data, thus there is no measured ground-truth of individual load energy consumption. However, from the labeled geometric and statistical events, it can be determined which estimates are significant underestimates or overestimates. The method from [23] underestimates the CPP energy consumption and overestimates the BP energy consumption. The rectangular method underestimates the energy of the BP. For the proposed method, the individual load energy consumption of the BP and CPP are close to the actual values.

A zoomed-in 35 minute window of the power stream is presented in order to provide intuition for these results. The measured aggregate power stream for this window with hand-labeled geometric and statistical events is shown in Fig. 12. The estimated power streams of the CPP and BP are shown in Fig. 13 using the proposed method, the method in [23], and rectangular integration. The method from [23], both with two and three states, cannot predict the low steady-state values that sometimes occur immediately after the BP turns on or before it turns off. For instance, in Fig. 13d between 46 and 51 minutes, there is an overestimation of power of the BP. By design, the method from [23] always has the correct total power at any time instant, and thus the correct total energy. However, the underestimation of power in one load leads to overestimation in power in another. In Fig. 13c, there are underestimates in power of the CPP in the periods that the BP power is overestimated. Using rectangular integration with the steady-state power calculated after the turn-on event means that the total energy may not be correct. Since this method does not assume a fixed number of devices, the BP underestimation (in Fig. 13f) does not affect the CPP estimation (in Fig. 13e). In contrast, the proposed method, as shown in Fig. 13a and Fig. 13b, is able to identify the statistical events and allocate the energy consumption to the correct load, as well as update the steady state after a statistical event region. Both the individual load energy consumption and the total energy consumption are close to the actual values.

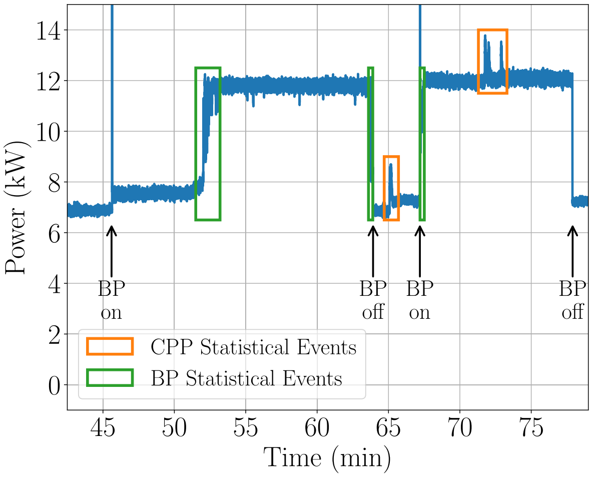
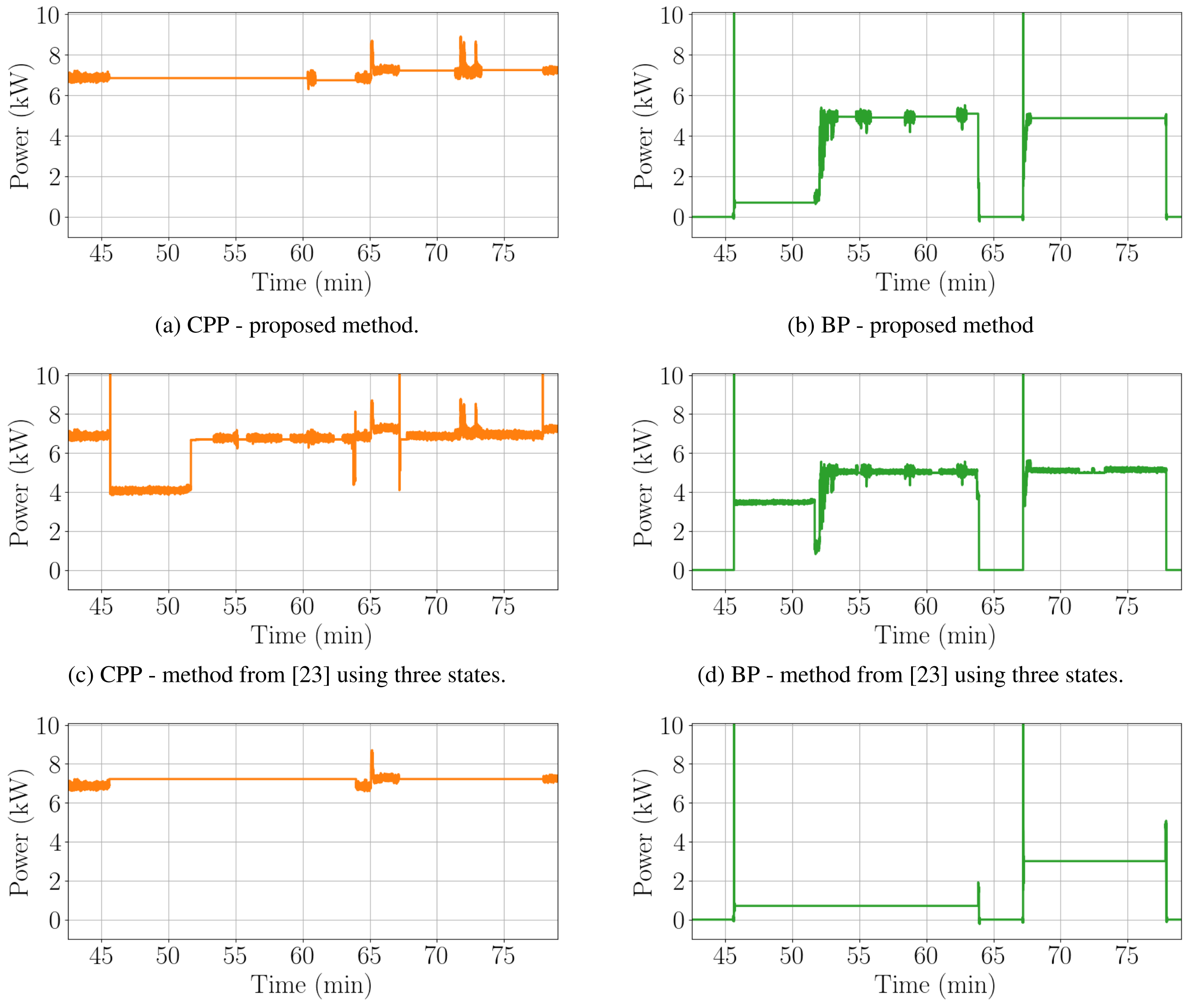


FIGURE 12. **Zoomed-in aggregate power stream with geometric and statistical events labeled.**

The tracked durations of CPP and BP statistical events in the two-hour window were 28.8 and 12.3 minutes, respectively. The duration of the CPP pump statistical events represents the working time of the relief valves in the hydraulic manifold. The relief valves regulate system pressure based on demand. Throttle commands that alter propeller blade pitch require greater system pressure and places a greater demand on the system. During periods of low demand with no changes in pitch, hydraulic oil interacts with and opens the low pressure relief valve. During periods of high demand, the hydraulic oil alternatively interacts with and opens the higher pressure relief valve; corresponding to the observed statistical events. Estimates of valve working time could aid in maintenance decisions. Currently, CPP hydraulic system relief values are tested on a fixed five-year cycle. BP statistical event detection could provide insight into pump health. With time, pump performance will inevitably deteriorate, for example, due to impeller wear. Decreased pump performance will cause changes in the time required to empty and fill storage tanks, and prime and clear air from the pump and associated piping. These changes will likely correspond to longer BP statistical events.

C. MACHINING EQUIPMENT

Monitoring the energy consumption of machining equipment is useful for efficiency, energy reduction, and condition-based maintenance in the manufacturing industry [18], [34]. This section uses a CNC router and an industrial bench grinder to demonstrate the utility of the proposed statistical event detector and energy disaggregation method. CNC machining is a manufacturing process where automated machines remove raw material with cutting tools. The power draw of CNC cutting machine operation varies based on cutting conditions such as the workpiece material, cutting speed, feed rate, and depth of cut [18]. Industrial grinders are used for sharpening cutting tools and shaping objects. The power draw of a grinder depends on the required load of the grinding operation.



(a) CPP - proposed method.

(b) BP - proposed method

(c) CPP - method from [23] using three states.

(d) BP - method from [23] using three states.

(e) CPP - rectangular integration.

(f) BP - rectangular integration.

FIGURE 13. Comparison of estimated power for the CPP and BP.

The CNC machine uses a Bosch 2.25 hp router motor, set to a fixed speed of 21,500 rpm. A CNC router was used to make a straight line cut through a piece of wood of various heights, as shown in Fig. 1. The major factors contributing to the total energy consumption are labeled on the power stream. There is an inrush up to 1700 W as the spindle accelerates (not fully shown). It then reaches a steady state of approximately 300 W while air-cutting. In this example, as the router cutting tool engages the wood, the power increases up to 500 W. A geometric event detector would likely only identify the spindle acceleration and deceleration as events and calculate the steady-state power for a short window before and after the event. However, this steady state would only correspond to air cutting, and not actual wood-cutting. The wood-cutting events and the large energy consumed during them would go undetected.

The industrial bench grinder used for testing was a Dayton 0.5 hp grinder with a six-inch grinding wheel, set to 1,800 rpm. An example power stream of machine operation is shown in Fig. 14. In this example, the startup and idle (baseline) operation is shaded blue, while the extra power required during grinding operation is shaded in gray. The base load of the unloaded grinding wheel (analogous to air-cutting of the CNC) is approximately 50 W. Grinding operations imitated burr removal and edge beveling in preparation for welding of mild steel flat stock. During grinding operation the power increased up to 300 W in this example. Integration of the shaded areas in Fig. 14 revealed that more than half of the total energy consumed in the example grinder stream is contained in the large fluctuations during loaded grinding operation (i.e., the area shaded gray). However, only the start and end of the base load operation would likely be

TABLE 3. Machining equipment statistical event classification.

		DNN			Proposed (ECDFs + k-NN)		
Equipment	**# of Windows**	$\mathbf{F_1}$ ($\sigma_{\mathbf{F1}}$)	**Pr** ($\sigma_{\mathbf{Pr}}$)	**Re** ($\sigma_{\mathbf{Re}}$)	$\mathbf{F_1}$ ($\sigma_{\mathbf{F1}}$)	**Pr** ($\sigma_{\mathbf{Pr}}$)	**Re** ($\sigma_{\mathbf{Re}}$)
CNC router	8	0.63 (0.25)	0.65 (0.23)	0.62 (0.27)	0.97 (0.03)	1.00 (0)	0.94 (0.06)
Grinder	28	0.91 (0.06)	0.90 (0.07)	0.91 (0.05)	0.99 (0.01)	0.98 (0.01)	1.00 (0)

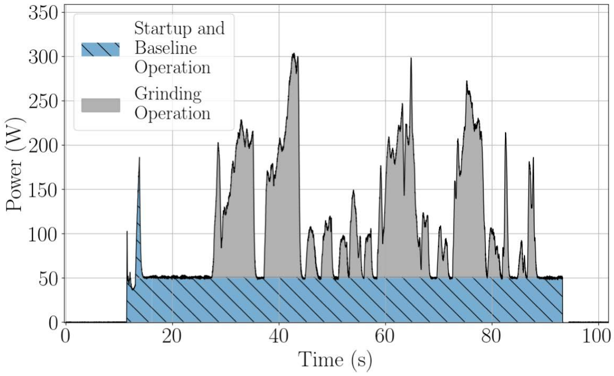


FIGURE 14. Power stream of industrial bench grinder.

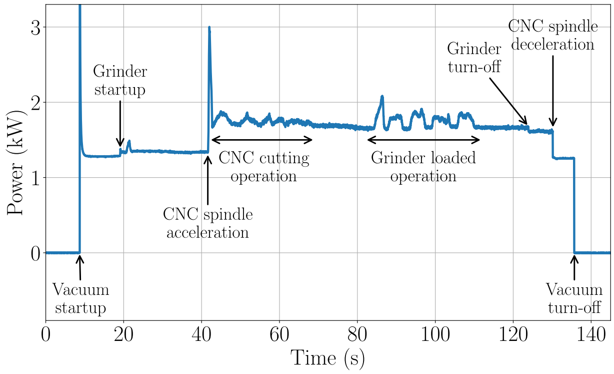
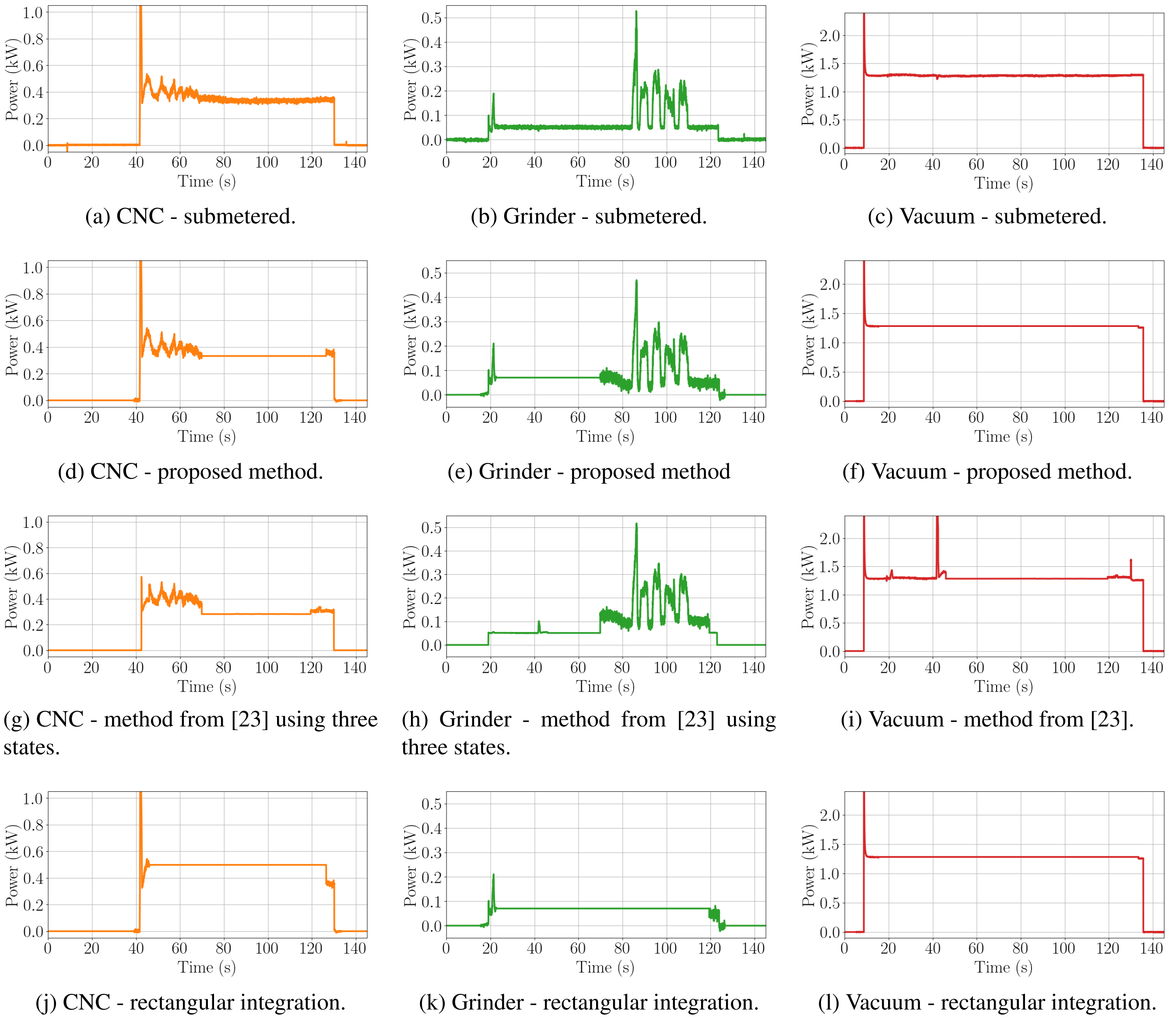


FIGURE 15. Ground truth aggregate power with labeled geometric and statistical events.

identified as events by a geometric edge detector. A majority of the energy consumed would be ''invisible'' using purely geometric event-based disaggregation.

To demonstrate the statistical classifier, the CNC and grinder were both run individually with several runs of normal cutting and grinding operation, respectively. Statistical windows were detected from the data streams using a rolling window and median filter both with length of 20 seconds. This window length and median filter length were chosen based on the duration of observed power fluctuations. The standard deviation threshold was set to $\sigma = 20$ W. This threshold was chosen to be larger than the standard deviation in the aggregate power stream during steady-state operation. In total, the dataset consisted of 21 and 68 windows of statistical activity for the CNC machine and grinder, respectively. These windows were randomly split into 60% training and 40% testing with data stratification. A k-NN classifier was trained with $k = 3$ without weighting. The dataset split and training was run 10 times for verification, with the results for the testing sets averaged and shown in Table 3. The high scores for the grinder and CNC can largely be explained by the behavior of reactive power and first-order difference of reactive power. The grinder appears capacitive and has negative reactive power during grinding operation. The CNC router appears inductive and has a positive reactive power which increases during cutting. As a result, the histograms of the normalized reactive power stream will skew right for the grinder and skew left for the CNC. The same DNN and training process as in Section IV-B was applied to this data, and the results are shown in Table 3. Although the DNN performed relatively well at identifying the grinder events, it struggled to identify CNC events. Similar to the DNN in the previous section, this is likely explained by a high input dimensionality and very small training dataset.

To demonstrate the statistical classifier incorporated with the energy estimation algorithm, the CNC and grinder were run in an aggregately monitored environment, which also included a 1.1 kW shop vacuum. The shop vacuum only has geometric events, and is therefore not included in Table 3. The standard deviation threshold was set to $\sigma = 20$ W. The aggregate power stream is shown in Fig. 15, with labels indicating main geometric and statistical events. Fig. 16 shows the estimated disaggregated power streams of the CNC, grinder, and vacuum, compared to the ground-truth submetered data and the estimates from the method in [23] and rectangular integration. For the CNC and grinder, the method in [23] was tested for the two-state (i.e, on, off) and three-state (i.e, on, loaded, off) scenarios. The vacuum was modelled as a two-state load in both scenarios. For the on state, the calculated mean values for the CNC, grinder, and vacuum were 289.0 W, 50.4 W, and 1280.6 W respectively, with standard deviations of 9.2 W, 1.0 W, and 5.7 W. For the loaded state, the mean values for the CNC and grinder were 372.8 W and 122.2 W, with standard deviations of 60.8 W and 66.8 W. Table 4 presents the resulting root-mean-square error (RMSE) for the power stream estimates compared to the ground truth submetered streams. The proposed method has the lowest RMSE for both the CNC and grinder. The proposed method and rectangular integration have the same RMSE for the vacuum, since the vacuum does not have any statistical event regions. Rectangular integration overestimated the energy of the CNC. Since the cutting operations started during the geometric window, the calculated steady state was high, as shown in Fig. 16j. At the same time, the missed energy during loaded grinder



(a) CNC - submetered.

(b) Grinder - submetered.

(c) Vacuum - submetered.

(d) CNC - proposed method.

(e) Grinder - proposed method

(f) Vacuum - proposed method.

(g) CNC - method from [23] using three states.

(h) Grinder - method from [23] using three states.

(i) Vacuum - method from [23].

(j) CNC - rectangular integration.

(k) Grinder - rectangular integration.

(l) Vacuum - rectangular integration.

FIGURE 16. Comparison of submetered and estimated power for the CNC, grinder, and vacuum (note that the y-axes for different loads are not the same).

operation led to underestimation of the energy of the grinder, as shown in Fig. 16k. For the method from [23], with only on and off states, the statistical events for a single load are divided between all the loads. The amount allocated to each load is based on the designated mean and variance of the steady-state power measurements and assuming a Gaussian distribution. That is, some of the grinding operation and some of the cutting operation were allocated to the incorrect load, leading to large RMSE values. When provided with the additional loaded state, this method can better allocate the energy during loaded operation to the correct load, as shown in Fig. 16g to 16i. However, there is still estimation error, since these regions are generally not Gaussian. The proposed method had accurate estimates for both steady-state and loaded operation, as shown in Fig. 16d to 16f.

The tracked durations of CNC and grinder statistical events were 23.9 and 49.6 seconds, respectively. This corresponds to the amounts of time that the CNC router bit was cutting and the grinding wheel was loaded. The methods described in this paper enable NILM systems to perform cumulative cutting time-based tool condition monitoring (TCM) of various machines from an aggregate point. Cumulative time TCM methods are commonly used to estimate cutting tool health and remaining useful life [35]. Non-uniform tool-life and loading conditions present challenges for cumulative time TCM. However, with the techniques presented, a NILM could ''weight'' cutting time by equipment power consumption and account for non-uniform wear or use power consumption as a proxy for tool condition.

TABLE 4. Machining loads power estimation RMSE.

	CNC	Grinder	Vacuum
Proposed	9.53 W	18.72 W	9.03 W
Method from [23] (2-state)	95.98 W	56.98 W	97.38 W
Method from [23] (3-state)	95.60 W	31.62 W	89.83 W
Rectangular	109.11 W	52.86 W	9.03 W

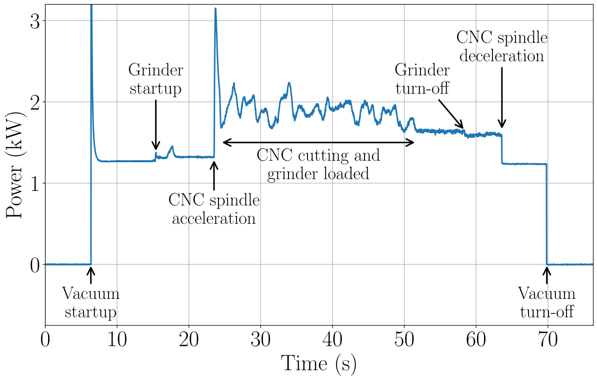


FIGURE 17. Ground truth aggregate power with labeled geometric and statistical events for CNC cutting and loaded grinder operation overlap case.

This work so far has assumed that statistical regions of multiple loads do not overlap. Although a rare scenario, the overlap condition could potentially be handled by adding classes representing each overlap condition. An overlap case was run in the submetered experiment in which the CNC is cutting and the grinder is loaded at the same time, shown in Fig. 17. As a preliminary attempt to identify overlapping statistical windows, a set of artificial overlapping windows was created by, for every pair of CNC and grinder training windows, taking the sum of the two windows. ECDFs were then generated for this overlap class. A testing set of ECDFs for the overlap case was generated by running the statistical event detector on the data in Fig. 17 as well as a subsequent overlapping run. A k-NN classifier was trained with $k = 3$ with a randomly split training set of CNC and grinder ECDFs and the artificial overlap ECDFs. It was then tested on the randomly split testing set of CNC and grinder ECDFs (4 and 14 testing examples, respectively) and the actual overlap ECDFs (with 10 testing examples). This was run 10 times, with average F_1 scores of 0.943, 0.989, and 0.968 for the CNC, grinder, and overlap case, respectively. Further work is required to investigate this and other overlap disaggregation techniques.

V. CONCLUSION

The results presented in this paper demonstrate the ability to disaggregate stochastic power behavior using statistical features in real time. Statistical events are distinct from the main load on- and off-events and provide indication of changing load demand. Tracking statistical events can create an "automatic logbook" of power system behavior that was previously invisible using conventional nonintrusive monitoring techniques. Statistical events generally relate to greater demand placed on the system. Thus, tracking these periods can aid in maintenance decisions. The presented case studies showcase the applicability of the proposed method in different industrial sectors. Future work includes developing techniques to further explore the disaggregation of individual energy present in overlapping statistical event regions. Extending the optimization technique in [23] to include non-Gaussian load power, particularly during regions of detected statistical region overlap, can be investigated.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support and dedication of the U.S. Coast Guard and in particular the crew of USCGC SPENCER for granting access to their ship.

REFERENCES

[1] Y. Yang, J. Zhong, W. Li, T. A. Gulliver, and S. Li, "Semisupervised multilabel deep learning based nonintrusive load monitoring in smart grids," *IEEE Trans. Ind. Informat.*, vol. 16, no. 11, pp. 6892–6902, Nov. 2020.

[2] C.-F. Lai, W.-C. Chien, L. T. Yang, and W. Qiang, "LSTM and edge computing for big data feature recognition of industrial electrical equipment," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2469–2477, Apr. 2019.

[3] W. Wichakool, Z. Remscrim, U. A. Orji, and S. B. Leeb, "Smart metering of variable power loads," *IEEE Trans. Smart Grid*, vol. 6, no. 1, pp. 189–198, Jan. 2015.

[4] P. Ducange, F. Marcelloni, and M. Antonelli, "A novel approach based on finite-state machines with fuzzy transitions for nonintrusive home appliance monitoring," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1185–1197, May 2014.

[5] F. M. Wittmann, J. C. López, and M. J. Rider, "Nonintrusive load monitoring algorithm using mixed-integer linear programming," *IEEE Trans. Consum. Electron.*, vol. 64, no. 2, pp. 180–187, May 2018.

[6] B. Buddhahai, W. Wongseree, and P. Rakkwamsuk, "An energy prediction approach for a nonintrusive load monitoring in home appliances," *IEEE Trans. Consum. Electron.*, vol. 66, no. 1, pp. 96–105, Feb. 2020.

[7] S. Zeinal-Kheiri, A. M. Shotorbani, and B. Mohammadi-Ivatloo, "Residential load disaggregation considering state transitions," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 743–753, Feb. 2020.

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

[9] W. Kong, Z. Y. Dong, D. J. Hill, J. Ma, J. H. Zhao, and F. J. Luo, "A hierarchical hidden Markov model framework for home appliance modeling," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 3079–3090, Jul. 2018.

[10] M. Z. A. Bhotto, S. Makonin, and I. V. Bajić, "Load disaggregation based on aided linear integer programming," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 64, no. 7, pp. 792–796, Jul. 2017.

[11] Y. Liu, J. Qiu, and J. Ma, "SAMNet: Toward latency-free non-intrusive load monitoring via multi-task deep learning," *IEEE Trans. Smart Grid*, vol. 13, no. 3, pp. 2412–2424, May 2022.

[12] F. Ciancetta, G. Bucci, E. Fiorucci, S. Mari, and A. Fioravanti, "A new convolutional neural network-based system for NILM applications," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–12, 2021.

[13] L. D. S. Nolasco, A. E. Lazzaretti, and B. M. Mulinari, "DeepDFML-NILM: A new CNN-based architecture for detection, feature extraction and multi-label classification in NILM signals," *IEEE Sensors J.*, vol. 22, no. 1, pp. 501–509, Jan. 2022.

[14] H. Hwang and S. Kang, "Nonintrusive load monitoring using an LSTM with feedback structure," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–11, 2022.

[15] K. Li, B. Yin, Z. Du, and Y. Sun, "A nonintrusive load identification model based on time-frequency features fusion," *IEEE Access*, vol. 9, pp. 1376–1387, 2021.

[16] G. Tang, K. Wu, and J. Lei, "A distributed and scalable approach to semi-intrusive load monitoring," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 6, pp. 1553–1565, Jun. 2016.

[17] Z. Xu, W. Chen, and Q. Wang, "A new non-intrusive load monitoring algorithm based on event matching," *IEEE Access*, vol. 7, pp. 55966–55973, 2019.

[18] J. Pan, C. Li, Y. Tang, W. Li, and X. Li, "Energy consumption prediction of a CNC machining process with incomplete data," *IEEE/CAA J. Automat. Sinica*, vol. 8, no. 5, pp. 987–1000, May 2021.

[19] A. K. Jain, R. P. W. Duin, and J. C. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, Jan. 2000.

[20] D. H. Green, S. R. Shaw, P. Lindahl, T. J. Kane, J. S. Donnal, and S. B. Leeb, "A MultiScale framework for nonintrusive load identification," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 992–1002, Feb. 2020.

[21] D. H. Green, "Electrical monitoring of electromechanical systems," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, May 2022.

[22] A. W. Langham, "Resolution tricks and disaggregation tools for smart power metering," M.S. thesis, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, May 2022.

[23] J. A. Mueller and J. W. Kimball, "Accurate energy use estimation for nonintrusive load monitoring in systems of known devices," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 2797–2808, Jul. 2018.

[24] D. H. Green, A. W. Langham, R. A. Agustin, D. W. Quinn, and S. B. Leeb, "Adaptation for automated drift detection in electromechanical machine monitoring," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 23, 2022, doi: 10.1109/TNNLS.2022.3184011.

[25] K. D. Lee, S. B. Leeb, L. K. Norford, P. R. Armstrong, J. Holloway, and S. R. Shaw, "Estimation of variable-speed-drive power consumption from harmonic content," *IEEE Trans. Energy Convers.*, vol. 20, no. 3, pp. 566–574, Sep. 2005.

[26] S. Ghosh and D. Chatterjee, "Artificial bee colony optimization based non-intrusive appliances load monitoring technique in a smart home," *IEEE Trans. Consum. Electron.*, vol. 67, no. 1, pp. 77–86, Feb. 2021.

[27] D. A. Darling, "The Kolmogorov–Smirnov, Cramer-von Mises tests," *Ann. Math. Statist.*, vol. 28, no. 4, pp. 823–838, Dec. 1957.

[28] O. Kramer, *Dimensionality Reduction with Unsupervised Nearest Neighbors*. Berlin, Germany: Springer, 2013, pp. 13–23.

[29] M. Simonov, G. Chicco, and G. Zanetto, "Event-driven energy metering: Principles and applications," *IEEE Trans. Ind. Appl.*, vol. 53, no. 4, pp. 3217–3227, Jul./Aug. 2017.

[30] T. Siauw and A. Bayen, *An Introduction to MATLAB Programming and Numerical Methods for Engineers*, 1st ed. New York, NY, USA: Academic, 2014.

[31] J. S. Donnal and S. B. Leeb, "Noncontact power meter," *IEEE Sensors J.*, vol. 15, no. 2, pp. 1161–1169, Feb. 2015.

[32] J. Paris, J. S. Donnal, Z. Remscrim, S. B. Leeb, and S. R. Shaw, "The sinefit spectral envelope preprocessor," *IEEE Sensors J.*, vol. 14, no. 12, pp. 4385–4394, Dec. 2014.

[33] T. J. Kane, "The NILM dashboard: Shipboard automatic watchstanding and real-time fault detection using non-intrusive load monitoring," Dept. Mech. Eng., Massachusetts Inst. Technol., Cambridge, MA, USA, Jun. 2019.

[34] H. S. Kang, J. Y. Lee, and D. Y. Lee, "An integrated energy data analytics approach for machine tools," *IEEE Access*, vol. 8, pp. 56124–56140, 2020.

[35] H. Liu and V. Makis, "Cutting-tool reliability assessment in variable machining conditions," *IEEE Trans. Rel.*, vol. 45, no. 4, pp. 573–581, Dec. 1996.

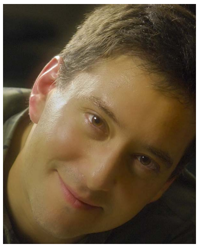
DAISY H. GREEN (Member, IEEE) received the B.S. degree in electrical engineering from the University of Hawai'i at Mānoa, Honolulu, HI, USA, in 2015, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 2018 and 2022, respectively. She is currently a Postdoctoral Associate with the Department of Architecture, MIT. Her research interests include signal processing for energy management and condition monitoring.

AARON W. LANGHAM (Graduate Student Member, IEEE) received the B.E.E. degree in electrical engineering from Auburn University, Auburn, AL, USA, in 2018, and the M.S. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2022, where he is currently pursuing the Ph.D. degree in electrical engineering and computer science. His research interests include signal processing and machine learning for energy management.

DEVIN W. QUINN received the M.S. degree in mechanical engineering from the Massachusetts Institute of Technology, in 2022. He was previously stationed as a Damage Control Officer aboard USCGC DILIGENCE and an Engineer Officer onboard USCGC ESCANABA. He is currently a Lieutenant Commander with the United States Coast Guard, Alameda, CA, USA.

THOMAS C. KRAUSE (Graduate Student Member, IEEE) received the B.S. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 2019, and the M.S. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2021, where he is currently pursuing the Ph.D. degree in electrical engineering and computer science.

STEVEN R. SHAW (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2000. He is currently a Professor with the Department of Electrical and Computer Engineering, Montana State University, Bozeman, MT, USA. His research interests include system identification and controls, instrumentation, and energy conversion.



STEVEN B. LEEB (Fellow, IEEE) received the Ph.D. degree from the Massachusetts Institute of Technology (MIT), in 1993. Since 1993, he has been a member at the Department of Electrical Engineering and Computer Science, MIT Faculty. He holds a joint appointment with the Department of Mechanical Engineering, MIT. His research interests include the development of signal processing algorithms for energy and real-time control applications.

• • •