IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 8, 2020, accepted May 21, 2020, date of publication May 27, 2020, date of current version June 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2997942

Residential Power Forecasting Based on Affinity Aggregation Spectral Clustering

CHINTHAKA DINESH, (Student Member, IEEE), STEPHEN MAKONIN, (Senior Member, IEEE), AND IVAN V. BAJIĆ, (Senior Member, IEEE)
School of Engineering Science, Simon Fraser University, BC V5A 1S6, Canada

Corresponding author: Chinthaka Dinesh (hchintha@sfu.ca)

This work was supported in part by the IC-IMPACTS, and in part by the NSERC Discovery Grants RGPIN-2018-06192 and RGPIN-2016-04590.

ABSTRACT Power utility companies rely on forecasting to anticipate future consumption needs, plan power production, and schedule the selling/purchasing of power. We present a novel method to forecast the power consumption of a single house based on non-intrusive load monitoring (NILM) and affinity aggregation spectral clustering, with the idea of extending it to forecasting consumption in a larger set of houses like a microgrid. First, we use a graph to model statistical relationships between appliances. Specifically, the ON/OFF time-of-day and state duration probabilities are used to compute graph edge weights and establish statistical relationships among appliances. Then, leveraging on our previous work on NILM, we disaggregate the smart meter aggregate power profile into individual appliance power profiles. With the disaggregated individual power profiles and the corresponding ON/OFF time-of-day and state duration probabilities, we next propose a method to forecast each appliance's power profiles using affinity aggregation spectral clustering. For the proposed method, we incorporate human behaviour and environmental influence in terms of calendar and seasonal contexts in order to enhance the forecasting performance. Finally, the results of appliance-level forecasting are aggregated to perform house-level forecasting. To test our proposed forecasting method, we use four publicly available datasets and compare our method against several existing approaches such as autoregressive integrated moving average, similar profile load forecast, artificial neural network, and recent NILM-based forecasting. Experimentally, we examine how well the proposed forecasting method can generalize appliance behaviours from one house to another. Results clearly show that our method is more accurate than existing approaches.

INDEX TERMS Appliance identification, demand side management, non-intrusive load monitoring, NILM, power forecasting, smart grid, spectral clustering.

I. INTRODUCTION

Forecasting is widely used by power utility companies to prepare for future consumption needs and optimize scheduling [1], [2]. Understanding future consumption needs allows the utility to plan power production, and in the cases where demand will be greater than supply, purchase additional power from other utility companies. Additionally, if there is a predicted excess in supply, then a utility may want to sell the excess power to others utilities in need. Without the ability to forecast, it is impossible to institute other schemes such as peak shaving or load shifting to prevent grid brownouts [3]. Additionally, accurate forecasting helps plan energy conservation programs. If households can reduce their energy consumption by an average of 14% then we can meet our COP21 Paris Climate Agreement goals for the household and commercial economic sector [4].

Most of the existing forecasting methods in the literature make predictions on the aggregate power signal directly, relying on the temporal dependence of the aggregate power signal. Such methods include estimating trends [5]–[7], fuzzy logic [8]–[10], Kalman filtering [8], [11], [12], support vector regression (SVR) [13], artificial neural networks (ANN) [8], [14]–[17], autoregressive integrated moving average (ARIMA) models [14], [18]–[20], similar profiles load forecast (SPLF) [21], [22], and the load fluctuation and

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

feature importance profile clustering [1]. According to the results presented in those papers, the aggregate power profiles of a very large number of houses (i.e. a power grid) typically have regular patterns that are more amenable to forecasting, so the accuracy of the above methods grows with the number of power profiles in the aggregate signal, as stated in [23]. However, forecasting accuracy at the level of microgrids or single houses is still a challenge.

The above forecasting methods are all based on predicting the aggregate power usage. An alternative strategy would be to disaggregate the power signal and apply prediction on individual components. In the econometric literature, such issues have been looked at in the context of macroeconomic forecasts, and the theory shows that under certain conditions, forecasting disaggregated components is more accurate than forecasting the aggregate [24]. The results we report in the present paper confirm this observation from econometrics for the case of power forecasting at the residential or microgrid level. The reason for this is that a stronger temporal dependence may exist in power signals of certain individual appliances, compared to the aggregate signal. This is easily seen in the case of cyclical appliances such as refrigerators, which turn ON and OFF roughly periodically. When the power signals of different appliances are aggregated, such temporal dependence may be disrupted, hence forecasting the aggregate power profile of a whole house may be less accurate than forecasting the power usage of individual appliances. For larger populations, at the level of the grid, aggregate forecasting, such as the methods listed above, is likely the only practical solution anyway, and it seems to work well in that case.

There is an added benefit to having any kind of individual appliance-level forecasting. Utility company initiatives such as *real-time demand side management* rely on appliance-level information to evaluate how much energy can be saved. Then, specific opt-in requests like unbalancing requests, load shedding, or energy price [25], [26] could to be sent to specific customers who could meet those requests. Additionally, *direct load control* becomes a feasible strategy for real-time demand side management [27]. Deferrable loads (such as HVAC systems, clothes dryers, and EV car charging stations) in opt-in houses can be identified through appliance-level forecasting and scheduled for automatic shutdown via utility remote control in real time.

Considering all of the above, a few residential power forecasting methods [2], [27]–[29] have been recently proposed, where the aggregate power signal is first decomposed into individual appliance signals via *non-intrusive load monitoring* (NILM),[1] then each appliance's power is forecasted separately, and finally, the total power forecast is formed by aggregating forecasted power levels of individual appliances. These existing NILM-based forecasting methods tend to be more accurate for residential level power forecasting than the traditional forecasting methods that use the aggregate signal [28]. However, all existing NILM-based forecasting methods [2], [27]–[29] assume individual appliance behaviour is uncorrelated with the behaviours of other appliances, which is not the case in practice [31]. As an alternative, we have recently proposed a new NILM-based forecasting method [29] based on graph spectral clustering (GSC), which incorporates appliance level correlations using ON/OFF state duration patterns. Hereafter, we refer to the method from [29] as GSC-NILM. In the present paper, we extend our previous work in [29] as described below.

Different from [29], in the present paper, mathematically, we capture appliance usage patterns into time-of-day and state duration probabilities, then affinity aggregation spectral clustering is used to perform the forecasting based on those probabilities. Human behaviours that affect residential power consumption are influenced by calendar context (working days, Saturdays, holidays) [22], [32], [33] and seasonal context (winter, spring, summer, fall) [29], [34]. In addition to the seasonal context used in [29], in this paper, we take both seasonal and calendar contexts into account in order to enhance the forecasting performance. Through extensive experiments we show that our proposed forecasting method offers higher accuracy compared to state-of-the-art forecasting methods based on GSC-NILM [29], NILM [27], ANN [14], ARIMA [14], and SPLF [22]. The main contributions of this paper are as follows:

1) We use a graph to model statistical relationships between appliances. Specifically, the ON/OFF time-of-day and state duration probabilities are used to compute graph edge weights and establish statistical relationships among appliances. This allows for powerful tools from graph theory and graph signal processing to be used in appliance behavior analysis and prediction.
2) We develop a new power forecasting method that combines NILM and affinity aggregation spectral clustering to improve residential power forecasting performance.
3) We incorporate human behaviour and environmental influence in terms of calendar and seasonal contexts into the proposed method in order to enhance the forecasting performance.
4) Experimentally, we examine how well the proposed forecasting method can generalize appliance behaviours from one house to another. For this purpose, we select common appliances from multiple houses, and perform training and testing on disjoint subsets of those houses.

The outline of the paper is as follows. In Section II, we briefly review affinity aggregation spectral clustering and the NILM method used in this paper. Next, a discussion of time-of-day and state duration probabilities is presented in Section III, followed by our proposed method in Section IV. Finally, experimental results and conclusions are presented in Sections V and VI, respectively. The main symbols and notation that will be used in this paper are as follows:

[1]NILM is a technique used to determine how much power each appliance is using from an aggregate power signal [30].

K The number of clusters
σ_k Affinity scaling factor for the k-th distance metric
$\mathbf{D}_k$ Distance matrix computed from the k-th distance metric
$\mathbf{A}_k$ Affinity matrix computed from the k-th distance metric
$\mathbf{A}$ Aggregate affinity matrix
$\mathbf{L}$ Laplacian matrix
$\mathbf{X}[i,:]$ i-th row of matrix $\mathbf{X}$
$\mathbf{X}[:,i\,]$ i-th column of matrix $\mathbf{X}$
$I_S(t)$ Indicator random variable for the set of appliances S, where S could be a single appliance a_i or a group of appliances. $I_S(t)$=1 means that all appliances in S are ON at time t.

ON-state
T_S duration of appliances in S

OFF-state
$\overline{T}_S$ duration of appliances in S

II. PRELIMINARIES

A. AFFINITY AGGREGATION SPECTRAL CLUSTERING

Spectral clustering is a graph-based unsupervised learning technique where data is embedded in a vector space of different dimension prior to being clustered. Affinity aggregation spectral clustering (AASC) [35], [36] is an extension of spectral clustering that uses more than one similarity (or distance) metric among data points. This form fits our purposes better, because it allows us to use both time-of-day and state duration probabilities in the forecasting process. AASC consists of three steps: graph construction, spectral embedding, and clustering.

1) GRAPH CONSTRUCTION

For a given set of n data points to be clustered, a fully-connected undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, is formed, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges. The weights of edges reflect the similarity between two nodes as a combination of different distance metrics between them. If there are L distance metrics used, there will be L affinity matrices $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_L \in \mathbb{R}^{n\times n}$, each defined as:

$$\mathbf{A}_k[i,j] = \mathbf{A}_k[j,i] = \begin{cases} \exp\left\{-\dfrac{\mathbf{D}_k[i,j]}{2\sigma_k^2}\right\}, & \text{if } i \neq j, \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

for $k = 1, 2, \ldots, L$, where $\mathbf{D}_k[i,j] = \mathbf{D}_k[j,i]$ is the distance between nodes i and j in terms of the k-th distance metric. Typically, these distance metrics are chosen in an application-specific manner. $\mathbf{A}_k[i,j] = \mathbf{A}_k[j,i]$ represent the ''similarity'' (or ''affinity'') between nodes i and j in terms of the k-th distance metric. Similarly to [37], in this paper we select σ_k in (1) as the standard deviation of the values in $\mathbf{D}_k$. Construction of $\mathbf{D}_k$'s will be presented in Section IV. Finally, the aggregate affinity matrix $\mathbf{A}$ is defined as:

$$\mathbf{A} = \sum_{k=1}^{L} \alpha_k \mathbf{A}_k, \tag{2}$$

where α_k is a weight associated with affinity matrix $\mathbf{A}_k$. These weights reflects the importance of various metrics for a particular application.

2) SPECTRAL EMBEDDING

The objective of this step is to map graph nodes into a vector space where high affinities translate into small Euclidean distances, so that one can run conventional clustering [35]. For this purpose, a new vector representation for each node is defined using the spectrum of the normalized Laplacian matrix $\mathbf{L}$ of the graph $\mathcal{G}$. First, $\mathbf{L} \in \mathbb{R}^{n\times n}$ is determined as [35]:

$$\mathbf{L} = \mathbf{W}^{-1/2}\mathbf{A}\mathbf{W}^{-1/2}, \tag{3}$$

where $\mathbf{W}$ is a diagonal matrix whose diagonal entries are summations of the corresponding columns of $\mathbf{A}$: $\mathbf{W}[i,i] = \sum_{j=1}^{n} \mathbf{A}[j,i]$. Then, eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m$ corresponding to the m largest eigenvalues of $\mathbf{L}$ are used to construct a matrix $\mathbf{X} \in \mathbb{R}^{n\times m}$, as $\mathbf{X}[:,i] = \mathbf{v}_i$. Finally, matrix $\mathbf{Y}$ is defined as

$$\mathbf{Y}[i,:] = \frac{\mathbf{X}[i,:]}{\|\mathbf{X}[i,:]\|_2} \tag{4}$$

where $\|\cdot\|_2$ is the ℓ_2 norm. Now, the i-th row of $\mathbf{Y}$, $\mathbf{Y}[i,:]$, represents node i in the $\mathbb{R}^m$ space [35], [37].

3) CLUSTERING

The rows of $\mathbf{Y}$ are treated as points in $\mathbb{R}^m$ and clustered using K-means clustering. Since $\mathbf{Y}[i,:]$ represents node i, a cluster of nearby (in Euclidean distance) rows of $\mathbf{Y}$ represents a cluster of nodes with similar affinities in the original node space. The number of clusters (K) is determined based on the eigenvalue difference distribution of the matrix $\mathbf{L}$ as follows [38]:

$$K = \arg\max_i(|\lambda_i - \lambda_{i+1}|), \tag{5}$$

where λ_i denotes the i-th largest eigenvalue of $\mathbf{L}$.

We treat individual appliances as nodes of a fully-connected undirected graph, and the aggregate affinity matrix $\mathbf{A}$ will represent correlation of appliance behaviours in terms of their time-of-day and state duration probabilities. Details are given in Sections III and IV.

B. NILM METHOD

NILM is a technique used to determine how much power each appliance is using from an aggregate power signal [30]. A literature review shows there are a number of recent NILM methods for residential appliance identification [34], [39]–[50]. The method in [39] is based on a super-state hidden Markov model to disaggregate multi-state loads. Load event matching-based methods are proposed

in [44], [45], [47] to perform load disaggregation. Moreover, several NILM algorithms were presented in [48], [49] based on deep neural networks. The external weather and environmental conditions were used to improve the NILM performance in [48]. Furthermore, generalized NILM algorithms across houses were proposed in [48], [50]. In [40], a NILM algorithm has been proposed based on aided linear integer programming. A temporal multi-label classification approach is used for load disaggregation in [41]. A graph signal processing (GSP)-based approach for NILM has been proposed in [43]. Recently, we have proposed a NILM method [34], which is an extension of [42], that takes appliance usage patterns such as time-of-day probabilities (Section III-A) into account. Incorporating time-of-day probabilities significantly improves NILM accuracy [42]. Therefore, we utilize [34] as the NILM method in the proposed forecasting approach. A brief review of [34] is given below.

The aggregate power signal is partitioned into non-overlapping observation windows (OWs), and in each OW, five principal components are extracted using Karhunen-Loeve Expansion (KLE). Algorithm 1 presents the main steps for a given OW. As seen in the algorithm, up to five iterations are executed, one for each principal component. In each iteration, two elimination steps are conducted, where each step removes some of the candidate sets of appliances. This is followed by the Maximum a Posteriori (MAP) estimation of the most likely set of appliances (S_{MAP}) that is turned ON in that OW.

Algorithm 1 Appliance Identification Algorithm From [42] for a Given OW

```
Set i = 1
Set execution = 1
Apply preprocessing;
while execution do
    Take the i-th principal component;
    Apply the two elimination steps;
    Conduct MAP estimation and find S_MAP;
    if P[I_S(t) = 1|Z_i] > 0.99 OR i == 5 then
        Output: S_MAP = S;
        Set execution = 0;
    else
        i = i + 1
    end if
end while
```

Let Z_i be the event that the first i principal components are those observed in the first i iterations in a given OW. MAP estimation finds the most likely set of appliances S that could have led to such an event. From the Bayes' rule,

$$P[I_S(t) = 1|Z_i] = \frac{P[Z_i|I_S(t) = 1] \cdot P[I_S(t) = 1]}{P[Z_i]}, \quad (6)$$

where t refers to the mid-point of the given OW. In the MAP estimation step, we consider only appliances sets S that have not been eliminated prior to the MAP step. Let us denote this set as $\mathcal{F} = \{S_1, S_2, \ldots, S_n\}$. The term $P[Z_i]$ in the denominator in (6) is common to all $S \in \mathcal{F}$, so removing it won't affect the maximization. Finally, the MAP problem becomes finding the set S that maximizes

$$P[I_S(t) = 1|Z_i] = P[Z_i|I_S(t) = 1] \cdot P[I_S(t) = 1]. \quad (7)$$

The term $P[Z_i|I_S(t) = 1]$ is computed in the last elimination step before the MAP estimation, as in [42]. while proxies for joint probabilities $P[I_S(t) = 1]$ are estimated following the procedure in [34] using time-of-day probabilities. Finally, the right-hand side of (7) can be computed for all sets $S \in \mathcal{F}$, and the set that maximizes it is selected as S_{MAP}. Appliances in this set are predicted to be ON at time t at the power levels associated with their states. Note that multi-state appliances are decomposed into multiple ON/OFF appliances with different ON-state power levels, as described in the next section.

As discussed in [51]–[53], personal data can be inferred from disaggregated load profiles. While privacy issues are somewhat orthogonal to the main objectives of the paper, in the case of the proposed NILM-based forecasting method, we feel that privacy is not as much of a concern. For example, when forecasting the power requirements of a group of houses, the proposed method disaggregates all appliances, but does not need to know which appliance belongs to which house.

III. TIME-OF-DAY AND STATE DURATION PROBABILITIES

There are three types of appliances that we encounter when disaggregating: binary state, multi-state, and continuous varying. A binary state appliance exhibits only one ON state and an OFF state (e.g., refrigerator). Multi-state appliances have an OFF state and several ON states, where each ON state represents a different power level (e.g., clothes dryer with two ON states – rotating drum and heating element). When modelling multi-state appliances, we create a separate binary state appliance for each ON state allowing us to simplify our model. Continuous varying appliances (e.g., variable speed motor) have no clear power-level state boundaries, so we first quantize them into $\mathcal{N}$ states following [39], and then create $\mathcal{N}-1$ virtual binary state appliances, as we did for multi-state appliances. Henceforth, we use "appliance" to mean binary state (ON/OFF) appliance, whether it is an actual binary state appliance or a virtual binary state appliance.

A. TIME-OF-DAY PROBABILITIES

Certain appliances have a higher chance of being used at certain times of day. For example, a toaster would often get used in the morning and likely never overnight. In this paper, we utilize such time-of-day ON patterns to forecast future power demand. Therefore, we compute a time-of-day probability for a set of appliances S as follows:

$$P[I_S(t) = 1] = \frac{m_S(t)}{N}, \quad (8)$$

which is the probability of all appliances in S being ON[2] at time t, where t is a certain time of day, N is the number of days in the training set, and $m_S(t)$ is the number of days in the training set in which all appliances in S were turned ON at time t. Here, S could be a single appliance ($S = \{a_i\}$), a pair of appliances ($S = \{a_i, a_j\}$), or a larger group of appliances ($S = \{a_{i_1}, a_{i_2}, \ldots, a_{i_n}\}$). For an illustrative example, $N = 4$ days are considered for training and typical ON/OFF status[3] of the appliance set S of these four days are shown in Fig. 1. According to the example, the appliance set S is ON three out of four training days at time t, so $m_S(t) = 3$, and hence $P[I_S(t) = 1] = 3/4 = 0.75$.

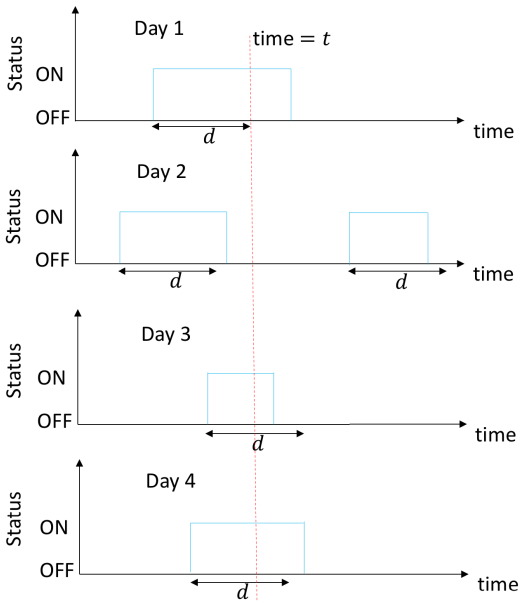
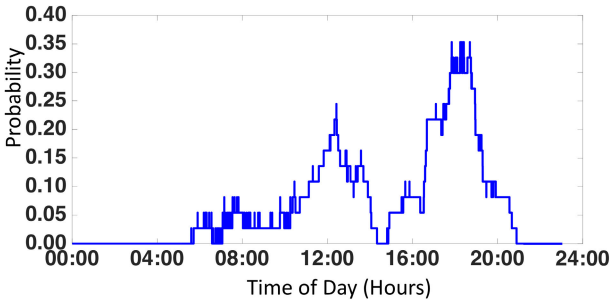
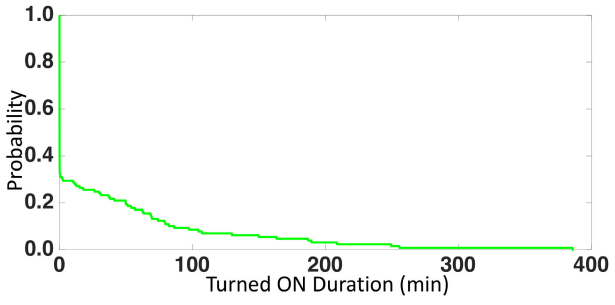


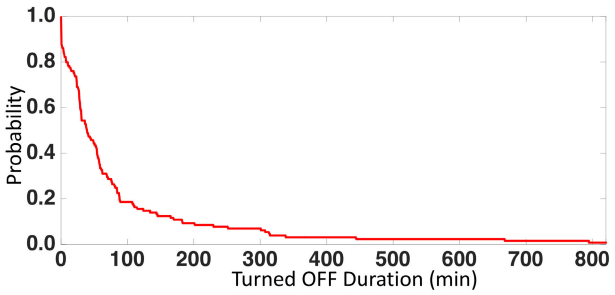
FIGURE 1. An example for ON/OFF status of an appliance set S for four days.



(a) Time-of-day for a desktop computer



(b) ON duration for a desktop computer



(c) OFF duration for a desktop computer

FIGURE 2. Example of time-of-day and ON/OFF duration probabilities for a desktop computer power trace found in tracebase [54].

Figure 2(a) shows time-of-day probability for a desktop computer. This example was computed from the fifty-day profiles in tracebase [54]. This figure shows that desktop computer clearly has a distinct time-of-day probability.

B. STATE DURATION PROBABILITIES

When appliances run (or turn ON) they tend to run for some time. For example, a washing machine often runs for between 30-60 minutes to wash a load of laundry, and an electric kettle takes a few minutes to boil a pot of water. We take advantage of this notion of expected ON duration patterns to forecast future power demand. For a given appliance a_i, the ON duration is defined as $T_{\{a_i\}}$ representing the appliance's elapsed ON time. A set of appliances S is said to be ON if and only if all the appliances in the set are ON. The ON duration of set S is defined as T_S and is the set's elapsed ON time during which all appliances are ON.

We treat state durations as random quantities whose probability distribution can be computed as:

$$P[T_S \geq d] = \frac{n_S^t}{n_S}, \tag{9}$$

where n_S is the number of times within the training set when all appliances in S are ON, and n_S^d is the number of times that all of them are ON for at least d time units.[4] According to the four training days shown in Fig. 1, the appliance set S is ON one time per day for day 1, day 3, and day 4, and it is ON two times in day 2. Then all together, the appliance set S is ON five times within the given four training days. Hence $n_S = 5$

[2]A set of appliances S is said to be ON if and only if all the appliances in the set are ON.

[3]Ideally, for any given appliance, OFF state corresponds to a zero power level and ON state corresponds to a considerable amount of power. However, due to measurement noise, OFF state may contain non-zero power. Moreover, in practice, the noise level is very small compared to the actual ON state power level [55]. Therefore, in this paper, if the measured power level of a given appliance at time t is bellow an empirically chosen threshold, then the status of the appliance at time t is considered as OFF, otherwise ON.

[4]In order to properly capture the ON duration probabilities even for appliances with very short ON duration, in practice, we set the minimum value of d as the sampling rate of the smart meter used to measure the power profile.

for this example. Further, as seen from Fig. 1, the appliance set S is ON three times for at least d time units, so $n_S^d = 3$, and hence $P[T_S \geq t] = 3/5 = 0.6$ for this example.

The set of appliances S is said to be OFF if at least one appliance in the set is OFF. OFF durations are also treated as random quantities with probability distribution computed as:

$$P[\overline{T}_S \geq d] = \frac{\overline{n}_S^d}{\overline{n}_S}, \tag{10}$$

where $\overline{T}_S$ is the set's elapsed OFF time, $\overline{n}_S$ is the number of times within the training set when S was OFF, and $\overline{n}_S^d$ is the number of times that S was OFF for at least d time units. The elapsed OFF time $\overline{T}_S$ is measured from the time instant when the first appliance in the set S turned OFF.

The ON duration probability distribution for a desktop computer is shown in Fig. 2(b) and its OFF duration probability distribution in Fig. 2(c). Fifty-day appliance traces in tracebase [54] were used to compute these probability distributions.

IV. PROPOSED METHOD

In this section we present the three main stages of our forecasting method: appliance state identification, ON-set prediction, and aggregation. To aid with the presentation, we define the following terms: t is the current time, t_p is previous time ($t_p < t$), t_f is future time ($t_f > t$), $d_c = t - t_p$ is the time elapsed between t_p and t, and $d_f = t_f - t_p$ is the time elapsed between t_p and t_f (see Fig. 3).

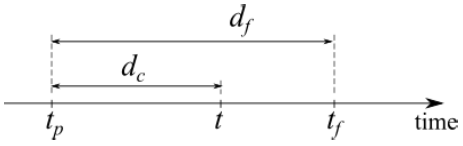


FIGURE 3. Illustration of t, t_p, t_f, d_c, and d_f.

A. APPLIANCE STATE IDENTIFICATION

Using the aggregated power signal of a house, ON/OFF states of appliances are identified at the current time t. Next, we identify the most recent start time instance of the current state of each appliance. We use one of our NILM methods [34], which is briefly discussed in Section II-B, to identify the state of each appliance at any given time up to the current time t. The disaggregated appliance state information is used with time-of-day probabilities (Section III-A) and state duration probabilities (Section III-B) to predict the ON state a future time t_f.

B. ON-SET PREDICTION

We define the ON-set as the set of all appliances that are ON at a particular time. We predict the ON-set at future time t_f with the help of AASC, which was reviewed in Section II-A.

1) GRAPH CONSTRUCTION

First, a fully connected undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, called an appliance graph is formed, where $\mathcal{V}$ is the set of all appliances (nodes of the graph) and $\mathcal{E}$ is the set of edges. An example of an appliance graph is shown in Fig. 4. Edge weights are aggregate affinities between the corresponding appliances. The aggregate affinities are derived from two distance metrics that are functions of appliances' joint time-of-day probabilities and state duration probabilities.

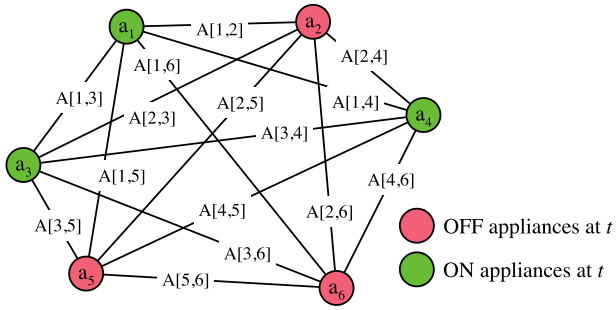


FIGURE 4. Example of an appliance graph.

The first distance metric is a function of appliances' joint time-of-day probabilities at a future time t_f. Specifically, for appliances a_i and a_j, we set the distance matrix $\mathbf{D}_1$ as:

$$\mathbf{D}_1[i,j] = \mathbf{D}_1[j,i] = 1 - P[I_{\{a_i,a_j\}}(t_f) = 1], \tag{11}$$

so that large joint time-of-day probability means small distance.

We choose the second distance metric as a function of appliances' joint state duration probabilities at a future time t_f. Suppose a pair of appliances $\{a_i, a_j\}$ in the graph is currently ON. Based on the history of appliance state identifications, we can determine the most recent time instant at which these two appliances turned ON. Let t_p be that time instant. Now consider the probability of these two appliances remaining in the ON state until the future time t_f or beyond. The conditional probability of this event, given that these two appliances are currently ON, is $P[(T_{\{a_i,a_j\}} \geq d_f)|(T_{\{a_i,a_j\}} \geq d_c)]$. Using Bayes' theorem,

$$P[(T_{\{a_i,a_j\}} \geq d_f)|(T_{\{a_i,a_j\}} \geq d_c)] = \frac{P[(T_{\{a_i,a_j\}} \geq d_c)|(T_{\{a_i,a_j\}} \geq d_f)] \cdot P[(T_{\{a_i,a_j\}} \geq d_f)]}{P[(T_{\{a_i,a_j\}} \geq d_c)]}. \tag{12}$$

From the definition of d_c and d_f, it is clear that

$$P[(T_{\{a_i,a_j\}} \geq d_c)|(T_{\{a_i,a_j\}} \geq d_f)] = 1,$$

so the required conditional probability becomes

$$P[(T_{\{a_i,a_j\}} \geq d_f)|(T_{\{a_i,a_j\}} \geq d_c)] = \frac{P[(T_{\{a_i,a_j\}} \geq d_f)]}{P[(T_{\{a_i,a_j\}} \geq d_c)]}. \tag{13}$$

The two terms on the right hand side of (13) are found from the state duration probabilities (Section III-B).

Finally, the distance $\mathbf{D}_2[i, j]$ between appliances a_i and a_j at time t_f is defined as

$$\mathbf{D}_2[i, j] = 1 - P[(T_{\{a_i,a_j\}} \geq d_f)|(T_{\{a_i,a_j\}} \geq d_c)]. \quad (14)$$

Having a pair of appliances $\{a_i, a_j\}$ that is currently OFF (as defined in Section III-B) means that either a_i or a_j or both are currently OFF. If the pair turned OFF at some past time t_p, we can use the same reasoning as above to compute the conditional probability of the pair staying OFF until a future time t_f as:

$$P[(\overline{T}_{\{a_i,a_j\}} \geq d_f)|(\overline{T}_{\{a_i,a_j\}} \geq d_c)] = \frac{P[(\overline{T}_{\{a_i,a_j\}} \geq d_f)]}{P[(\overline{T}_{\{a_i,a_j\}} \geq d_c)]}, \quad (15)$$

where the right-hand side probabilities are found from the state duration probabilities (Section III-B). The distance $\mathbf{D}_2[i, j]$ between appliances a_i and a_j at future time t_f is defined as:

$$\mathbf{D}_2[i, j] = P[(\overline{T}_{\{a_i,a_j\}} \geq d_f)|(\overline{T}_{\{a_i,a_j\}} \geq d_c)]. \quad (16)$$

Having obtained $\mathbf{D}_1$ and $\mathbf{D}_2$, affinity matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ are constructed from (1). Next, the aggregate affinity matrix $\mathbf{A}$ is obtained from (2), where $\alpha_1 = \alpha_2 = 1$ in this work.

2) SPECTRAL EMBEDDING AND CLUSTERING

After constructing the appliance graph at future time t_f, spectral embedding and clustering discussed in Section II-A are performed. The spectral representation of the graph (matrix $\mathbf{Y}$ in Section II-A) is obtained as in (3)-(4), where m is found by eigen-gap analysis similar to (5).

Matrix $\mathbf{Y}$ is a spectral representation of the appliance graph, where appliance a_i is represented as a point in $\mathbb{R}^m$ by the i-th row of $\mathbf{Y}$, i.e. $\mathbf{Y}[i, :]$. Distances (in $\mathbb{R}^m$) between different rows of $\mathbf{Y}$ reflect "affinities" between appliances from matrix $\mathbf{A}$, which are based on probabilities of corresponding appliances being ON at the same time. The smaller the Euclidean distance between two rows, the higher the affinity between the corresponding appliances, which means the higher the chance they are ON at the same time. In essence, matrix $\mathbf{Y}$ allows us to use Euclidean distance-based methods such as K-means clustering [56], where we would otherwise have to use computations that involve joint probabilities. Following (5), eigenvalue differences of the matrix $\mathbf{L}$ determine the number of clusters (K) where the j-th cluster represents appliance set S_j. For each cluster the average Euclidean distance AED_j of the vectors associated with S_j is:

$$AED_j = \frac{1}{N_j} \sum_{a_i \in S_j} \|\mathbf{Y}[i, :] - \mathbf{c}_j\|_2, \quad (17)$$

where N_j is the number of appliances in S_j, and $\mathbf{c}_j$ is the centroid of the vectors representing those appliances: $\mathbf{c}_j = \frac{1}{N_j} \sum_{a_i \in S_j} \mathbf{Y}[i, :]$.

As discussed in [29], [34], the value of AED_j is inversely related to the joint probability of appliances in S_j being ON at future time t_f. However, for clusters that contain a single appliance, $AED_j = 0$. After clustering, if any singleton clusters (e.g., $S_j = \{a_l\}$) exist, we add a dummy appliance $a_{l'}$ to S_j such that $S_j = \{a_l, a_{l'}\}$. Now, $\mathbf{D}_1[l, l']$ and $\mathbf{D}_2[l, l']$ at future time t_f are computed as:

$$\mathbf{D}_1[l, l'] = 1 - P[I_{\{a_l\}}(t_f) = 1], \quad (18)$$

$$\mathbf{D}_2[l, l'] = \begin{cases} 1 - P[(T_{\{a_l\}} \geq d_f)|(T_{\{a_l\}} \geq d_c)], & \text{if } a_l \text{ is ON at } t \\ P[(\overline{T}_{\{a_l\}} \geq d_f)|(\overline{T}_{\{a_l\}} \geq d_c)], & \text{if } a_l \text{ is OFF at } t, \end{cases} \quad (19)$$

where $\mathbf{D}[i, l'] = \mathbf{D}[i, l]$ and $\mathbf{D}[l', i] = \mathbf{D}[l, i]$ for $i \neq l$. Once dummy appliances have been added to singleton clusters a new graph is constructed, new spectral embedding is created, and new AED_j's are computed. Finally, the appliance set with the smallest AED_j is predicted to be ON at future time t_f. Our ON-set prediction method is summarized in Fig. 5.

C. AGGREGATION

The mean power level of each appliance is obtained as in [42]. The total forecasted power level at future time t_f is the summation of the mean power level of each appliance in the predicted ON-set. We can then scale up our method to forecast the power demand of a set of houses, or a microgrid, by the summation of the forecasted powers for each house. A summary of the whole forcasting algorithm for a microgrid with M houses (denoted $h_1, \ldots, h_M$) to predict the total power demand at a future time t_f is given in Algorithm 2.

D. CALENDAR CONTEXT AND SEASONAL CONTEXT

Human behaviours that affect residential power consumption are influenced by *calendar context* – working days[5] (WD),

[5]Days from Monday to Friday without special holidays.

Algorithm 2 The Whole Forcasting Algorithm for a Microgrid to Predict the Total Power Demand at a Given Future Time t_f

1: **for** $i = 1 : M$ **do**
2: Identify the current status of each appliance in house h_i at the current time t (as discussed in Section IV-A);
3: Identify the most recent start-time instance of the current state of each appliance in h_i (as discussed in Section IV-A);
4: Predict ON-set of appliances in h_i at time t_f by following the flowchart in Fig. 5 (as discussed in Section IV-B);
5: Compute the total forecasted power level in h_i (denoted as p_h_i) at t_f (as discussed in Section IV-C);
6: **end for**
7: Compute the total power demand p of the microgrid at t_f as $p = \sum_{i=1}^{M} p\_h_i$;

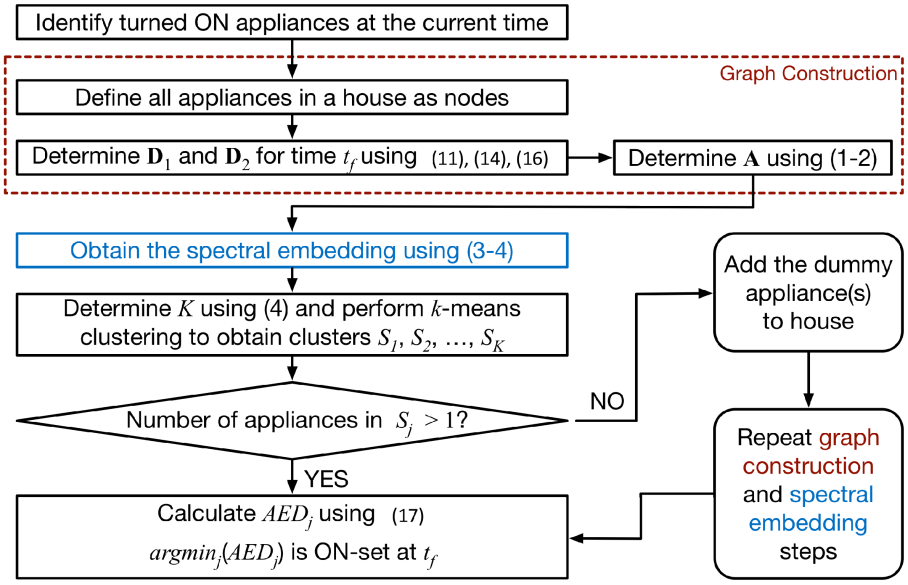


FIGURE 5. Prediction of the ON-set at a given future time t_f.

Saturdays[6] (SD), holidays[7] (HD) – [22], [32], [33] and *seasonal context* – winter, spring, summer, fall – [29], [34]. For example, during a holiday, people are likely to watch TV for longer than on a working day; in summer, people are likely to use a fan for longer than in winter. In order to take such human behaviours into account, we extend the forecasting model (including the NILM method discussed in Section II-B) to include *seasonal and calendar context*. A regular context-free model is trained over the whole dataset, independent of the season and calendar, while seasonal and calendar context-based models are trained separately on each season (winter, spring, summer, fall) and each calendar context (WD, SD, HD). When both contexts are used, we get a fairly refined information about the likely behaviour (e.g., working days in winter, holidays in summer, etc.). As shown in the next section (section V-B), this type of context information improves the forecasting accuracy.

V. EXPERIMENTAL RESULTS

To measure how well our forecasting method works, we compare against the following state-of-art forecasting methods: GSC-NILM [29], NILM [27], ANN [14], ARIMA [14], and SPLF [22]. Experiments were ran using MATLAB R2015b (without code optimization) on a 2.2 GHz MacBook Pro with Intel core i7 processor and 16GB memory.

We use the Mean Absolute Percentage Error (MAPE) to measure prediction accuracy:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y(t_i) - \widehat{y}(t_i)|}{y(t_i)} \times 100\%, \tag{20}$$

[6]All Saturdays without holidays
[7]all Sundays and special holidays (Christmas, New Year's day. etc.)

where n is the number of time samples, $\widehat{y}(t_i)$ is the forecasted power level at time t_i, and $y(t_i)$ is the ground truth power level at time t_i. MAPE is a widely used performance metric. However, it becomes unstable when $y(t_i)$ values approach zero. To supplement it, we also use Root Mean Square Error (RMSE):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y(t_i) - \widehat{y}(t_i))^2}. \tag{21}$$

Data from publicly available datasets (REDD [57], Rainforest Automation Energy Dataset (RAE) [58], Almanac of Minutely Power dataset version 2 (AMPds2) [55], and *tracebase* [54]) was used to test the performance in four case studies described below. In order to have a consistent sampling time interval across datasets, REDD, RAE, and *tracebase* data was down-sampled to 1-minute intervals to match the sampling interval in AMPds2. Twenty appliances out of 122 were selected from *tracebase*, as shown in Table 1.

TABLE 1. Appliances selected from *tracebase*.

Appliance Types	Selected Appliances
Single State	60 W and 100 W lamps, 1800 W and 2000 W water kettles, microwave oven, toaster, iron, video projector
Continuous Varying	LCD television, CRT television, CRT monitor, TFT monitor, remote desktop, desktop computer
Multiple States	cooking stove, refrigerator, washing machine, dish washer, laundry dryer, washer dryer combo

For each case study, we use inputs for training and testing from one of the above datasets. In terms of training, to "train" a forecaster means computing ON/OFF time-of-day and state duration probabilities in the proposed method, computing ON/OFF duration probabilities in [29], determining ON/OFF

TABLE 2. MAPE(%) (left sub-column) and RMSE(kW) (right sub-column) results for houses in REDD and RAE using 180-minute ahead forecast.

House Name	Proposed		GSC-NILM [29]		NILM [27]		ANN [14]		ARIMA [14]		SPLF [22]	
REDD 1	**3.88**	**0.47**	4.69	0.57	8.14	1.08	9.21	1.17	9.63	1.29	10.1	1.48
REDD 2	**3.37**	**0.43**	4.28	0.51	7.87	1.10	9.51	1.24	9.39	1.27	9.42	1.21
REDD 3	**5.12**	**0.57**	6.34	0.71	14.7	1.41	16.5	1.65	17.7	1.69	18.5	1.52
REDD 4	**5.49**	**0.61**	6.16	0.68	13.0	1.31	15.9	1.62	16.3	1.63	18.9	1.66
REDD 5	**7.27**	**0.65**	8.04	0.72	15.5	1.43	19.5	1.79	19.5	1.82	20.6	2.01
REDD 6	**6.41**	**0.62**	7.24	0.69	13.2	1.46	20.0	1.85	21.4	1.92	20.6	1.98
RAE	**3.75**	**0.44**	4.32	0.55	7.92	1.02	9.08	1.12	8.99	1.21	9.36	1.17

duration patterns of individual appliance for the method in [27], and obtaining model parameters for the two methods in [14]. Moreover, aggregate power profiles for previous n days are used as training inputs for methods ANN [14], ARIMA [14], and SPLF [22]. However, individual power profiles for previous n days are used as training inputs for the methods NILM [27] and GSC_NILM [29], and the method proposed in this paper. In addition, for the context-based versions of the six methods, calendar and seasonal contexts of those n days are used as training inputs for all six methods.

In terms of testing, the current aggregate power profile (i.e. power signal from the current time to one day before the current time) is used as the test input for all six forecasting methods. For context-based versions of the six methods, calendar and seasonal context corresponding to the time t_f are also given as inputs for all methods.

A. CASE STUDY 1

A total of seven houses were used in this case study – six from REDD and one from RAE. For each REDD house, the first 26 days of active power profiles were used for the training, allowing us to use the next 30 days for testing. For the RAE house, the first 25 days were used for training and the next 38 days for testing. During training we computed the time-of-day probabilities in (8) and ON/OFF-state duration probabilities in (9)-(10) for our method. We also determined the ON/OFF duration patterns of individual appliances for the method in [27], and obtained model parameters for the two methods in [14]. For [22], the training set represents the search space of similar profiles.

We performed a 180-minute ahead forecast using 1-minute increments with our method and existing methods [14], [22], [27], [29]. We measured the performance of each method using MAPE and RMSE metrics. The results are shown in Table 2, with the most accurate predictions indicated in bold. The results show that our proposed method greatly outperforms existing methods [14], [22], [27] on all houses, with a 44-72% reduction in MAPE and a 47-74% reduction in RMSE. Moreover, the GSC-NILM in [29] is the next most accurate forecasting method, still better than the existing methods. Comparing the full and simplified version of our method, we see that using both state duration and time-of-day probabilities offers better accuracy, with MAPE reduced by 9-21% and RMSE reduced by 10-20% compared to using state duration probabilities only.

TABLE 3. Forecasting average execution time (s) for houses in REDD and RAE using 180-minute ahead forecast.

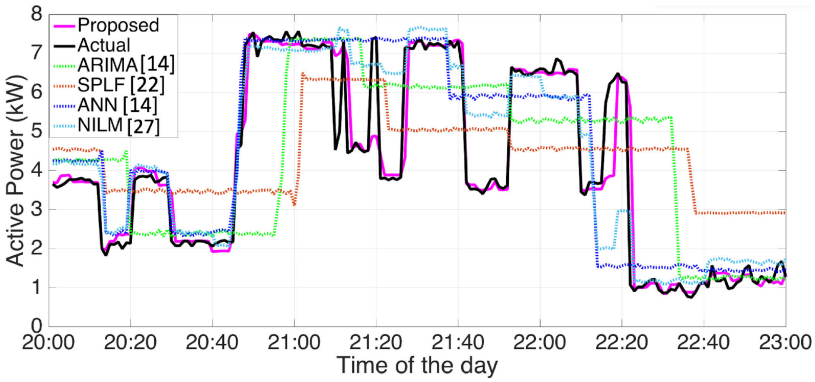
House Name	Proposed	NILM [27]	ANN [14]	ARIMA [14]	SPLF [22]
REDD 1	1.07	0.91	0.84	3.72	0.93
REDD 2	1.06	0.93	0.78	3.11	1.08
REDD 3	1.24	1.01	0.86	2.99	1.07
REDD 4	1.11	0.95	0.79	3.01	0.92
REDD 5	1.18	0.97	0.91	3.04	1.13
REDD 6	1.07	0.91	0.86	2.97	1.08
RAE	1.08	0.87	0.81	3.18	1.05

TABLE 4. t-test results for Case Study 1.

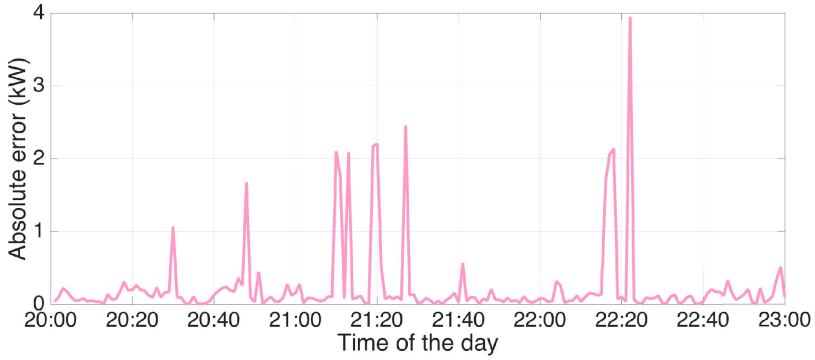
House Name	p value	
	for APE	for SE
REDD1	3.74×10^{-4}	2.01×10^{-6}
REDD2	$1.87 \times 10{-}5$	$2.23 \times 10{-}6$
REDD3	$2.77 \times 10{-}7$	1.07×10^{-8}
REDD4	5.25×10^{-3}	2.05×10^{-4}
REDD5	3.57×10^{-2}	3.18×10^{-3}
REDD6	2.14×10^{-3}	3.05×10^{-4}
RAE	1.14×10^{-3}	2.42×10^{-4}

MAPE is an average absolute percentage error (APE) per sample, while RMSE is the square root of the average squared error (SE) per sample. Hence, MAPE is a sample average while RMSE is derived from the sample average. We ask the question whether the observed difference in these sample averages between different methods is statistically significant? We use a two-sample t-test with unknown variance [59] to answer this question. Specifically, we compare the set of APE (SE) samples of the proposed method and APE (SE) samples of the next-best method (with next lowest MAPE (RMSE)) using the aforementioned t-test, with the null hypothesis that both sets of samples come from distributions with the same mean and unknown variance. All the p-values related to this case study are reported in Table 4. Typically, $p < 0.05$ is taken as an indication of statistically significant difference. As seen in the table, all p-values are less than 10^{-3}, indicating that the errors produced by the proposed method are indeed significantly lower than those of the next-best method (and thereby other methods as well).

A sample of forecasting results for REDD House 1 is shown in Fig. 6a, where it is clear that our method tends to be more accurate than the alternative methods. The existing forecasting methods are generally able to predict upward and downward swings in power consumption. However, predictions of power levels is less accurate, often with prediction



(a) Forecasting power profiles for the given actual power profile using various forecasting methods including the proposed one



(b) Absolute error of the proposed forecasting signal as a function of time

FIGURE 6. A sample of forecasting results for REDD House 1.

lag or lead. Moreover, Fig. 6a and 6b indicate that relatively large errors are made (by our method, as well as other methods) near transients. We focus on errors made by the proposed method, as the most accurate one among the tested methods. The distribution of large errors ($>$ 1kW) relative to the nearest transient for all test signals used in this case study is shown in Fig. 7. As the figure shows, over 70% of large errors occur within a minute of a transient, and over 90% of large errors occur within 5 minutes of a transient. This shows that, indeed, transients represent a challenge for the forecasting methods, including ours.

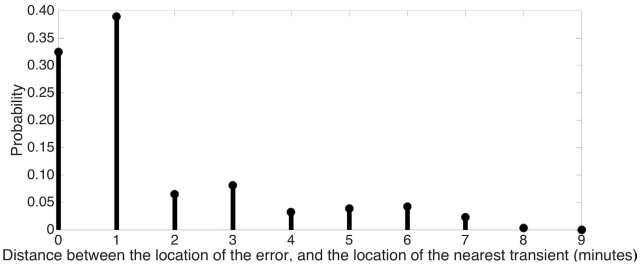


FIGURE 7. Histogram of the distances between the location of the error and the location of the nearest transient.

In terms of average execution time shown in Table 3, the ANN forecasting method [14] runs fastest, seconded by the NILM forecasting method [27], and followed closely by the SPLF forecasting method [22] and our method. The ARIMA forecasting method [14] is about three times slower, on average.

B. CASE STUDY 2

Our goal in this case study is to evaluate the effectiveness of using seasonal and calendar context in power forecasting on the AMPds2 house. AMPds2 contains two years of data sampled at 1-minute intervals. We train on the first year and test on the second year. We train each method in four ways (Section IV-D): (1) context-free (CF); (2) using seasonal context (SC); (3) using calendar context (CC); and (4) using both SC and CC.

We performed a similar 180-minute ahead forecast using 1-minute increments as we did in Case Study 1. We measured the accuracy of each forecast for each CC per season from April 2013 to March 2014 in AMPds2. Results are presented in Table 5 using MAPE and RMSE metrics. Comparing against existing methods [14], [22], [27] our method performs significantly better, having MAPE reduced by 51-73% and RMSE reduced by 49-67% using a CF model. As seen in the results, both SC and CC are able to bring improvements to the corresponding CF model, for all forecasting methods. Specifically, CC is slightly more accurate than CF, while SC

TABLE 5. MAPE(%) (top sub-row) and RMSE(kW) (bottom sub-row) results for a 180-minute ahead power forecasting for the AMPds2 house.

Season & Calendar condition	Proposed				NILM [27]				ANN [14]				ARIMA [14]				SPLF [22]			
	CF	SC	CC	SC + CC	CF	SC	CC	SC + CC	CF	SC	CC	SC + CC	CF	SC	CC	SC + CC	CF	SC	CC	SC + CC
Winter & WD	4.03	3.17	3.64	**3.12**	8.53	8.00	8.27	7.88	10.52	9.94	10.11	9.83	11.32	11.91	11.02	10.77	11.28	10.65	10.71	10.54
	0.60	0.44	0.48	**0.42**	1.21	1.18	1.19	1.15	1.32	1.24	1.25	1.22	1.42	1.34	1.34	1.31	1.37	1.34	1.34	1.31
Winter & SD	4.10	3.24	3.55	**3.20**	8.58	7.98	8.26	7.95	10.47	10.08	10.17	9.91	11.35	10.85	11.07	10.63	11.22	10.64	10.67	10.52
	0.62	0.48	0.50	**0.44**	1.23	1.20	1.20	1.17	1.31	1.28	1.30	1.23	1.44	1.32	1.37	1.30	1.40	1.33	1.34	1.30
Winter & HD	4.02	3.14	3.28	**3.11**	8.48	8.01	8.18	7.91	10.42	9.85	9.97	9.77	11.24	10.93	10.98	10.80	11.35	10.74	10.73	10.61
	0.58	0.44	0.45	**0.41**	1.19	1.16	1.17	1.14	1.27	1.24	1.25	1.19	1.42	1.34	1.33	1.32	1.38	1.33	1.35	1.32
Spring & WD	3.32	3.04	3.14	**2.93**	7.94	7.28	7.43	7.14	9.48	9.11	9.08	9.05	9.53	9.08	9.14	9.01	10.87	10.29	10.34	10.18
	0.51	0.47	0.49	**0.38**	1.14	1.05	1.11	1.03	1.24	1.17	1.19	1.14	1.38	1.22	1.26	1.20	1.35	1.25	1.26	1.22
Spring & SD	3.27	2.96	3.07	**2.93**	8.03	7.22	7.47	7.17	9.52	9.15	9.10	9.07	9.55	9.12	9.17	9.06	10.82	10.34	10.32	10.21
	0.50	0.47	0.46	**0.40**	1.15	1.08	1.12	1.05	1.25	1.17	1.20	1.15	1.38	1.23	1.28	1.21	1.33	1.24	1.25	1.20
Spring & HD	3.31	3.02	3.01	**2.95**	7.96	7.20	7.38	7.18	9.46	9.12	9.10	9.05	9.54	9.10	9.14	9.02	10.92	10.47	10.43	10.36
	0.51	0.47	0.44	**0.39**	1.17	1.07	1.10	1.06	1.23	1.15	1.17	1.12	1.35	1.27	1.25	1.20	1.37	1.33	1.31	1.25
Summer & WD	3.21	2.63	2.94	**2.50**	9.13	8.08	8.21	8.01	11.48	10.78	10.83	10.61	11.75	11.04	11.12	10.93	12.51	11.94	12.07	11.81
	0.40	0.33	0.36	**0.28**	1.20	1.10	1.12	1.07	1.35	1.23	1.26	1.21	1.31	1.27	1.29	1.25	1.36	1.29	1.30	1.25
Summer & SD	3.28	2.65	2.83	**2.53**	9.14	8.11	8.17	8.07	11.54	10.81	10.87	10.69	11.69	11.01	11.07	10.87	12.46	11.92	11.97	11.75
	0.43	0.35	0.36	**0.31**	1.18	1.10	1.10	1.08	1.32	1.20	1.21	1.17	1.30	1.26	1.26	1.24	1.34	1.28	1.30	1.26
Summer & HD	3.25	2.61	2.77	**2.48**	9.18	8.14	8.18	8.06	11.46	10.73	10.75	10.62	11.73	11.14	11.15	11.03	12.55	12.03	12.05	11.87
	0.41	0.34	0.35	**0.28**	1.17	1.10	1.11	1.08	1.37	1.28	1.30	1.26	1.35	1.30	1.32	1.27	1.40	1.32	1.31	1.28
Fall & WD	3.71	2.84	2.91	**2.72**	8.76	7.85	7.93	7.81	10.15	9.64	9.71	9.52	12.06	11.47	11.56	11.36	12.24	11.60	11.68	11.51
	0.50	0.36	0.40	**0.33**	1.15	1.08	1.09	1.05	1.35	1.20	1.22	1.15	1.45	1.31	1.33	1.27	1.32	1.28	1.30	1.25
Fall & SD	3.68	2.87	2.91	**2.81**	8.74	7.88	7.95	7.84	10.07	9.77	9.74	9.60	12.08	11.51	11.50	11.38	12.22	11.97	11.98	11.45
	0.49	0.38	0.40	**0.34**	1.12	1.08	1.09	1.02	1.33	1.22	1.26	1.20	1.43	1.32	1.35	1.26	1.35	1.29	1.28	1.24
Fall & HD	3.85	2.91	2.87	**2.69**	8.75	7.90	7.89	7.80	10.04	9.67	9.72	9.50	12.16	11.63	11.71	11.47	12.25	12.03	12.05	11.65
	0.52	0.40	0.40	**0.32**	1.17	1.11	1.09	1.07	1.38	1.24	1.25	1.18	1.50	1.35	1.37	1.29	1.36	1.28	1.28	1.27

TABLE 6. Forecasting accuracy and average execution time for a 400-house community in Case Study 3.

	Proposed	NILM [27]	Micro-Forecast			Macro-Forecast		
			ANN [14]	ARIMA [14]	SPLF [22]	ANN [14]	ARIMA [14]	SPLF [22]
MAPE (%)	**1.71**	4.52	5.27	6.08	5.35	5.82	6.85	5.78
RMSE (kW)	**32**	81	99	127	117	124	138	123
Average execution time (s)	1.61	1.37	**0.85**	3.47	1.68	0.83	3.31	1.63

is considerably more accurate than CF. However, the best accuracy (indicated in bold) is achieved when both SC and CC are used (SC+CC in the table). Under SC+CC model, the proposed method still outperforms all others with MAPE reduced by 52-76% and RMSE reduced by 57-75%.

C. CASE STUDY 3

Next, we evaluate the performance of aggregated power forecasting for a larger set of houses, like a microgrid. The datasets used previously only contain data for a small number of houses, which means that we need to create a data for a large number of houses. Using appliance traces in tracebase, we created a large set of "virtual" houses. The appliances we selected from tracebase (see table 1) have power traces greater than 120 days, allowing us to use the first 50 days for training. Each virtual house contained 12 out of 20 randomly chosen appliances. For each appliance chosen, a ten consecutive day power trace was randomly selected to represent that house's appliance ten days of usage. This allowed us generate ten-day power traces for all 400 virtual houses.

Again, we performed a similar 180-minute ahead forecast using 1-minute increments for all 400 houses. Results are reported in Table 6 using MAPE and RMSE metrics. Two sets of results for the methods in [14], [22] are shown: (1) we forecast each house separately then sum to get the aggregate forecast (called "micro-forecast"), and (2) forecasting the aggregate directly (called "macro-forecast"). Results show that our method significantly outperforms the other methods in all cases. MAPE is reduced by 63% and RMSE is reduced by 58% compared to the best alternative, which is the NILM forecasting method [27].

In terms of the average execution time per sample to predict 180-minute ahead (Table 6) there are some things to note. Given that there are 400 houses, we need to forecast each house separately and sum to get the total forecasted power. The forecasting of each house is done independently which means that we can forecast houses in parallel – similar to a map-reduce problem. Therefore, we record the maximum execution time for a given prediction across all 400 houses and calculate the average maxima over all predictions. The resulting average is reported as the average execution time[8] in the table. Again, we found the ANN forecasting method [14] to be the fastest, but our method still has a practical and viable average execution time, and is much more accurate.

[8]There are 400 houses in case study 3. For the proposed method, [27], and "aggregating forecast" version of [14], [22], we need to calculate the forecasted power level for each house separately and obtain their summations to calculate the total forecasted power. In practice, the forecast for each house can be performed in parallel. Therefore, similar to our recent work on [29], we measure the maximum execution time for a given prediction (180 minutes ahead) across all 400 houses and calculate the average of these maximuma over all predictions. This average is reported in the table.

TABLE 7. The MAPE(%)/RMSE(kW) for a 180-minute ahead power forecasting of the REDD Houses 2, 3, and 5 using REDD House 1 for training.

Testing Houses	Proposed		NILM [27]		ANN [14]		ARIMA [14]		SPLF [22]	
REDD 2	**7.14**	**1.24**	12.8	1.51	15.3	1.51	17.8	1.78	18.5	1.84
REDD 3	**10.8**	**1.36**	19.1	1.92	22.4	2.07	23.8	2.18	25.1	2.21
REDD 5	**12.7**	**1.43**	20.1	2.03	24.5	2.26	25.2	2.34	26.7	2.41

TABLE 8. The MAPE(%)/RMSE(kW) for a 180-minute ahead power forecasting of the REDD Houses 1, 3, and 5 using REDD House 2 for training.

Testing Houses	Proposed		NILM [27]		ANN [14]		ARIMA [14]		SPLF [22]	
REDD 1	**7.51**	**1.22**	13.2	1.54	15.1	1.64	16.7	1.77	19.2	1.88
REDD 3	**11.2**	**1.38**	19.1	1.94	21.5	2.04	23.2	2.14	24.7	2.18
REDD 5	**13.1**	**1.48**	20.4	2.10	24.1	2.27	26.1	2.41	26.9	2.47

TABLE 9. The MAPE(%)/RMSE(kW) for a 180-minute ahead power forecasting of the REDD Houses 1, 2, and 5 using REDD House 3 for training.

Testing Houses	Proposed		NILM [27]		ANN [14]		ARIMA [14]		SPLF [22]	
REDD 1	**8.12**	**1.27**	14.1	1.60	15.7	1.68	17.3	1.81	19.8	1.87
REDD 2	**10.8**	**1.34**	18.3	1.86	22.1	1.97	23.5	2.11	25.2	2.21
REDD 5	**13.6**	**1.44**	20.7	2.06	23.8	2.25	25.9	2.38	27.7	2.48

TABLE 10. The MAPE(%)/RMSE(kW) for a 180-minute ahead power forecasting of the REDD Houses 1, 2, and 3 using REDD House 5 for training.

Testing Houses	Proposed		NILM [27]		ANN [14]		ARIMA [14]		SPLF [22]	
REDD 1	**8.38**	**1.20**	14.8	1.57	17.1	1.64	18.2	1.86	19.5	1.93
REDD 2	**10.3**	**1.36**	17.4	1.83	23.8	1.91	23.7	2.14	24.3	2.26
REDD 3	**12.8**	**1.48**	21.3	2.10	23.4	2.18	25.1	2.42	25.8	2.45

D. CASE STUDY 4

In this case study, we examine how well the forecasting methods can generalize. We select four houses (Houses 1, 2, 3, and 5) in REDD, which have six common appliances: microwave oven, refrigerator, dish washer, washer dryer, lamp, and water kettle. The first 26 days of active power profiles were used for training. For common appliances, data from one of the houses was used for training. This training house changes in different test cases (Tables 7-10). The non-common appliances were trained separately in each house. The next 30 days of the remaining three houses were used for testing.

Power consumption was predicted 180-minutes ahead in 1-minute steps using our method as well as the methods in [14], [22], [27]. The MAPE and RMSE results are shown in Tables 7-10 when the training house rotates among REDD Houses 1, 2, 3, and 5. The best results are indicated in bold. As seen in the tables, our method outperforms others by a considerable margin, with MAPE reduced by 21-44% and RMSE reduced by 18-41%.

Comparing the results in Tables 7-10 with the corresponding results in Table 2, we see that the accuracies of all methods have dropped compared to those in Table 2. This is no surprise – the models are now asked to generalize appliance behaviour from one house to other houses. Nonetheless, the forecast accuracy of our method in this scenario is still better than the accuracy of other methods even when they are trained and tested on the data from the same house (Table 2).

Similar to Case Study 1, we performed two-sample t-tests with unknown variance for all other case studies in this section. All p-values were less than 10^{-2}, indicating that the forecasting errors in the above tables produced by the proposed method are significantly lower than those of the next best method, and thereby other methods as well.

VI. CONCLUSION

We proposed a novel forecasting method that exploits correlation among appliance usage patterns through joint state duration and time-of-day probabilities. Current and previous appliance states are identified and used to perform the power forecast for each appliance. The forecasting results of each appliance are summed up to produce the aggregated power forecast. The proposed method is applicable to power forecasting for a single house, or a group of houses, such as a microgrid. Our method was tested using four publicly available datasets and compared against five state-of-the-art forecasting methods from the literature. Superior accuracy was achieved in each case. The proposed framework allows for seasonal and calendar context-based forecasting and the results show that both contexts clearly help improve the forecasting accuracy. Further, we demonstrated that our method has better generalization ability than other methods, by training on appliance data from one house and testing on data from other houses. However, relatively large errors are made by all forecasting methods, including our own, near transients of the power signal. Therefore, in the future, we aim to extend the proposed method to improve the forecasting performance near transients.

REFERENCES

[1] N. Huang, W. Wang, S. Wang, J. Wang, G. Cai, and L. Zhang, "Incorporating load fluctuation in feature importance profile clustering for day-ahead aggregated residential load forecasting," *IEEE Access*, vol. 8, pp. 25198–25209, 2020.

[2] K. Park, S. Yoon, and E. Hwang, "Hybrid load forecasting for mixed-use complex based on the characteristic load decomposition by pilot signals," *IEEE Access*, vol. 7, pp. 12297–12306, 2019.

[3] V. Zdraveski, M. Todorovski, D. Trajanov, and L. Kocarev, "Dynamic load balancing and reactive power compensation switch embedded in power meters," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 64, no. 4, pp. 422–426, Apr. 2017.

[4] A. Rodriguez-Silva and S. Makonin, "Universal non-intrusive load monitoring (UNILM) using filter pipelines, probabilistic knapsack, and labelled partition maps," in *Proc. IEEE PES Asia–Pacific Power Energy Eng. Conf. (APPEEC)*, Dec. 2019, pp. 1–6.

[5] W. Feng, X. Kai, Y. Erkeng, L. Guoqi, and W. Manyi, "A simple and effective ultrashort term load forecasting method," *Power Syst. Technol.*, vol. 20, no. 3, pp. 41–43 and 48, 1996.

[6] D. S. Luo and H. Y. He, "A shape similarity criterion based curve fitting algorithm and its application in ultra-short-term load forecasting," *Power Syst. Technol.*, vol. 31, no. 21, pp. 81–84, 2007.

[7] Z. L. Yang, G. Q. Tang, Y. M. Song, and R. Z. Cao, "Improved cluster analysis based ultra-short term load forecasting method," *Autom. Electr. Power Syst.*, vol. 29, no. 24, pp. 83–86, 2005.

[8] C. Guan, P. B. Luh, L. D. Michel, and Z. Chi, "Hybrid Kalman filters for very short-term load forecasting and prediction interval estimation," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 3806–3817, Nov. 2013.

[9] A. Tale, A. S. Gusain, J. Baguli, R. Sheikh, and A. Badar, "Study of load forecasting techniques using fuzzy logic," *Int. J. Adv. Res. Electr., Electron. Instrum. Eng.*, vol. 6, no. 2, pp. 512–561, Feb. 2017.

[10] H. H. Çevik and M. Çunkas, "A fuzzy logic based short term load forecast for the holidays," *Int. J. Mach. Learn. Comput.*, vol. 6, no. 1, p. 57, 2016.

[11] D. J. Trudnowski, W. L. McReynolds, and J. M. Johnson, "Real-time very short-term load prediction for power-system automatic generation control," *IEEE Trans. Control Syst. Technol.*, vol. 9, no. 2, pp. 254–260, Mar. 2001.

[12] C. Lynch, M. J. O'Mahony, and R. A. Guinee, "Electrical load forecasting using an expanded Kalman filter bank methodology," *IFAC-PapersOnLine*, vol. 49, no. 25, pp. 358–365, 2016.

[13] Y. Peng, Y. Wang, X. Lu, H. Li, D. Shi, Z. Wang, and J. Li, "Short-term load forecasting at different aggregation levels with predictability analysis," in *Proc. IEEE Innov. Smart Grid Technol. Asia (ISGT Asia)*, May 2019, pp. 3385–3390.

[14] A. Marinescu, C. Harris, I. Dusparic, S. Clarke, and V. Cahill, "Residential electrical demand forecasting in very small scale: An evaluation of forecasting methods," in *Proc. 2nd Int. Workshop Softw. Eng. Challenges Smart Grid (SE4SG)*, May 2013, pp. 25–32.

[15] W. Kong, Z. Y. Dong, D. J. Hill, F. Luo, and Y. Xu, "Short-term residential load forecasting based on resident behaviour learning," *IEEE Trans. Power Syst.*, vol. 33, no. 1, pp. 1087–1088, Jan. 2018.

[16] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on LSTM recurrent neural network," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 841–851, Jan. 2019.

[17] S.-V. Oprea and A. Bara, "Machine learning algorithms for short-term load forecast in residential buildings using smart meters, sensors and big data solutions," *IEEE Access*, vol. 7, pp. 177874–177889, 2019.

[18] B. Stephen, X. Tang, P. R. Harvey, S. Galloway, and K. I. Jennett, "Incorporating practice theory in sub-profile models for short term aggregated residential load forecasting," *IEEE Trans. Smart Grid*, vol. 8, no. 4, pp. 1591–1598, Jul. 2017.

[19] D. Alberg and M. Last, "Short-term load forecasting in smart meters with sliding window-based ARIMA algorithms," *Vietnam J. Comput. Sci.*, vol. 5, nos. 3–4, pp. 241–249, Jun. 2018.

[20] T.-H. Dang-Ha, F. M. Bianchi, and R. Olsson, "Local short term electricity load forecasting: Automatic approaches," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 4267–4274.

[21] E. Paparoditis and T. Sapatinas, "Short-term load forecasting: The similar shape functional time-series predictor," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 3818–3825, Nov. 2013.

[22] M. Tucci, E. Crisostomi, G. Giunta, and M. Raugi, "A multi-objective method for short-term load forecasting in European countries," *IEEE Trans. Power Syst.*, vol. 31, no. 5, pp. 3537–3547, Sep. 2016.

[23] A. F. Ebrahim and O. A. Mohammed, "Pre-processing of energy demand disaggregation based data mining techniques for household load demand forecasting," *Inventions*, vol. 3, no. 3, pp. 2411–5134, 2018.

[24] D. F. Hendry and K. Hubrich, "Combining disaggregate forecasts or combining disaggregate information to forecast an aggregate," Eur. Central Bank, Working Paper 1155, Feb. 2010. [Online]. Available: https://www.ecb.europa.eu/pub/pdf/scpwps/ecbwp1155.pdf

[25] K. Basu, V. Debusschere, and S. Bacha, "Residential appliance identification and future usage prediction from smart meter," in *Proc. 39th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Nov. 2013, pp. 4994–4999.

[26] F. Ye, Y. Qian, and R. Q. Hu, "A real-time information based demand-side management system in smart grid," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 2, pp. 329–339, Feb. 2016.

[27] S. Welikala, C. Dinesh, M. P. B. Ekanayake, R. I. Godaliyadda, and J. Ekanayake, "Incorporating appliance usage patterns for non-intrusive load monitoring and load forecasting," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 448–461, Jan. 2019.

[28] M. Wurm and V. C. Coroamă, "Poster abstract: Grid-level short-term load forecasting based on disaggregated smart meter data," *Comput. Sci. Res. Develop.*, vol. 33, nos. 1–2, pp. 265–266, Feb. 2018.

[29] C. Dinesh, S. Makonin, and I. V. Bajić, "Residential power forecasting using load identification and graph spectral clustering," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 66, no. 11, pp. 1900–1904, Nov. 2019.

[30] G. W. Hart, "Nonintrusive appliance load monitoring," *Proc. IEEE*, vol. 80, no. 12, pp. 1870–1891, Dec. 1992.

[31] K. Basu, V. Debusschere, and S. Bacha, "Appliance usage prediction using a time series based classification approach," in *Proc. 38th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2012, pp. 1217–1222.

[32] G. Chicco, R. Napoli, and F. Piglione, "Load pattern clustering for short-term load forecasting of anomalous days," in *Proc. IEEE Porto Power Tech*, vol. 2, Sep. 2001, p. 6.

[33] L. Semeraro, E. Crisostomi, A. Franco, A. Landi, M. Raugi, M. Tucci, and G. Giunta, "Electrical load clustering: The Italian case," in *Proc. IEEE PES Innov. Smart Grid Technol., Eur.*, Oct. 2014, pp. 1–6.

[34] C. Dinesh, S. Makonin, and I. V. Bajic, "Incorporating time-of-day usage patterns into non-intrusive load monitoring," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Nov. 2017, pp. 1–5.

[35] A. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. NIPS*, 2002, pp. 849–856.

[36] H.-C. Huang, Y.-Y. Chuang, and C.-S. Chen, "Affinity aggregation for spectral clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 773–780.

[37] W.-M. Song, T. Di Matteo, and T. Aste, "Hierarchical information clustering by means of topologically embedded graphs," *PLoS ONE*, vol. 7, no. 3, Mar. 2012, Art. no. e31929.

[38] U. von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, Dec. 2007.

[39] 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.

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

[41] K. Basu, V. Debusschere, S. Bacha, U. Maulik, and S. Bondyopadhyay, "Nonintrusive load monitoring: A temporal multilabel classification approach," *IEEE Trans. Ind. Informat.*, vol. 11, no. 1, pp. 262–270, Feb. 2015.

[42] C. Dinesh, B. W. Nettasinghe, R. I. Godaliyadda, M. P. B. Ekanayake, J. Ekanayake, and J. V. Wijayakulasooriya, "Residential appliance identification based on spectral information of low frequency smart meter measurements," *IEEE Trans. Smart Grid*, vol. 7, no. 6, pp. 2781–2792, Nov. 2016.

[43] K. He, L. Stankovic, J. Liao, and V. Stankovic, "Non-intrusive load disaggregation using graph signal processing," *IEEE Trans. Smart Grid*, vol. 9, no. 3, pp. 1739–1747, May 2018.

[44] Y. Xiao, Y. Hu, H. He, D. Zhou, Y. Zhao, and W. Hu, "Non-intrusive load identification method based on improved KM algorithm," *IEEE Access*, vol. 7, pp. 151368–151377, 2019.

[45] 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.

[46] B. Zhao, L. Stankovic, and V. Stankovic, "On a training-less solution for non-intrusive appliance load monitoring using graph signal processing," *IEEE Access*, vol. 4, pp. 1784–1799, 2016.

[47] H. Liu, Q. Zou, and Z. Zhang, "Energy disaggregation of appliances consumptions using HAM approach," *IEEE Access*, vol. 7, pp. 185977–185990, 2019.

[48] M. Kaselimi, N. Doulamis, A. Voulodimos, E. Protopapadakis, and A. Doulamis, "Context aware energy disaggregation using adaptive bidirectional LSTM models," *IEEE Trans. Smart Grid*, early access, Feb. 17, 2020, doi: 10.1109/TSG.2020.2974347.
[49] K. Chen, Y. Zhang, Q. Wang, J. Hu, H. Fan, and J. He, "Scale- and context-aware convolutional non-intrusive load monitoring," *IEEE Trans. Power Syst.*, vol. 35, no. 3, pp. 2362–2373, May 2020.
[50] N. Batra, A. Singh, and K. Whitehouse, "Neighbourhood NILM: A big-data approach to household energy disaggregation," 2015, *arXiv:1511.02900*. [Online]. Available: http://arxiv.org/abs/1511.02900
[51] M. A. Lisovich, D. K. Mulligan, and S. B. Wicker, "Inferring personal information from demand-response systems," *IEEE Secur. Privacy Mag.*, vol. 8, no. 1, pp. 11–20, Jan. 2010.
[52] G. Eibl and D. Engel, "Influence of data granularity on smart meter privacy," *IEEE Trans. Smart Grid*, vol. 6, no. 2, pp. 930–939, Mar. 2015.
[53] S. Makonin, L. Guzman Flores, R. Gill, R. A. Clapp, L. Bartram, and B. Gill, "A consumer bill of rights for energy conservation," in *Proc. IEEE Canada Int. Humanitarian Technol. Conf. (IHTC)*, Jun. 2014, pp. 1–6.
[54] A. Reinhardt, P. Baumann, D. Burgstahler, M. Hollick, H. Chonov, M. Werner, and R. Steinmetz, "On the accuracy of appliance identification based on distributed load metering data," in *Proc. Sustain. Internet ICT Sustain. (SustainIT)*, Oct. 2012, pp. 1–9.
[55] S. Makonin, B. Ellert, I. V. Bajić, and F. Popowich, "Electricity, water, and natural gas consumption of a residential house in canada from 2012 to 2014," *Sci. Data*, vol. 3, no. 1, pp. 1–12, Dec. 2016.
[56] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, Jul. 2002.
[57] J. Z. Kolter and M. J. Johnson, "REDD: A public data set for energy disaggregation research," in *Proc. ACM SustKDD*, 2011, pp. 59–62.
[58] 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, Feb. 2018.
[59] D. C. Montgomery, G. C. Runger, and N. F. Hubele, *Engineering Statistics*. Hoboken, NJ, USA: Wiley, 2009.

CHINTHAKA DINESH (Student Member, IEEE) received the B.Sc. Eng. degree in electrical & electronic engineering from the University of Peradeniya, Peradeniya, Sri Lanka, in 2012. He is currently pursuing the Ph.D. degree with the School of Engineering Science, Simon Fraser University (SFU), Burnaby, Canada. He was a Research Assistant at Iterative and Digital Media Institute (IDMI), National University of Singapore (NUS), from 2013 to 2014. Moreover, he was a Temporary Lecturer with the Department of Electrical and Electronics Engineering, University of Peradeniya, from 2014 to 2016. During his PhD, he has been working as a Visiting Research Student with the National Institute of Informatics (NII), Tokyo, Japan, from March 2018 to July 2018, and the Department of Electrical Engineering and Computer Science, York University, Toronto, Canada, from August 2019 to January 2020. His current research interests include graph signal processing, smart grid and load monitoring, convex optimization, and 3D Point Clouds. He has received number of awards and scholarships from SFU during his PhD. He was a recipient of the IEEE SPS Student Travel Grant for ICIP 2019. Moreover, one of his article was selected as a Spotlight Paper (top 10% accepted papers) in ICIP 2019.

STEPHEN MAKONIN (Senior Member, IEEE) received the Ph.D. degree in computing science with Simon Fraser University, in 2014, in the area of computational sustainability. He is currently an Adjunct Professor in Engineering Science and the Principal Investigator of the Computational Sustainability Lab, Simon Fraser University (SFU). He has been a Software Engineer for over 24 years working for various local/international industry clients. He is currently a registered Professional Engineering (P.Eng.) with Engineers and Geoscientists BC. His research interests include computational sustainability and the understanding of socioeconomic issues that pertain to technological advancement. He is an expert in data engineering, software engineering, and a world-renowned researcher in non-intrusive load monitoring (NILM) and disaggregation. He is currently the Vice-Chair of the IEEE Signal Processing Society Vancouver Chapter. He also serves as an Editorial Board Member of *Nature's Scientific Data Journal*.

IVAN V. BAJIĆ (Senior Member, IEEE) is currently a Professor of Engineering Science and the Co-Director of the Multimedia Lab, Simon Fraser University, Burnaby, BC, Canada. His research interests include signal processing and machine learning with applications to multimedia processing, ergonomics, compression, and communications. He has authored about a dozen and coauthored another ten dozen publications in these fields. His articles have won awards at the IEEE ICME 2012 and the IEEE ICIP 2019, and other (e.g., top n %) recognitions at ICME, ICIP and CVPR. He was the Chair of the Media Streaming Interest Group of the IEEE Multimedia Communications Technical Committee, from 2010 to 2012. He is currently an Elected Member of the IEEE Multimedia Signal Processing Technical Committee and the IEEE Multimedia Systems and Applications Technical Committee. He has served on the organizing and/or program committees of the main conferences in the field, and has received five reviewing awards, most recently at IEEE ICASSP 2019. He was an Associate Editor of IEEE TRANSACTIONS ON MULTIMEDIA AND the *IEEE Signal Processing Magazine*. He is also an Area Editor of *Signal Processing: Image Communication* and a Senior Area Editor of the IEEE SIGNAL PROCESSING LETTERS.