Energy Disaggregation via Deep Temporal Dictionary Learning

Mahdi Khodayar, *Student Member, IEEE*, Jianhui Wang, *Senior Member, IEEE*, and Zhaoyu Wang, *Member, IEEE*

***Abstract*— This paper presents a novel nonlinear dictionary learning (DL) model to address the energy disaggregation (ED) problem, i.e., decomposing the electricity signal of a home to its operating devices. First, ED is modeled as a new temporal DL problem where a set of dictionary atoms is learned to capture the most representative temporal features of electricity signals. The sparse codes corresponding to these atoms show the contribution of each device in the total electricity consumption. To learn powerful atoms, a novel deep temporal DL (DTDL) model is proposed that computes complex nonlinear dictionaries in the latent space of a long short-term memory autoencoder (LSTM-AE). While the LSTM-AE captures the deep temporal manifold of electricity signals, the DTDL model finds the most representative atoms inside this manifold. To simultaneously optimize the dictionary and the deep temporal manifold, a new optimization algorithm is proposed that alternates between finding the optimal LSTM-AE and the optimal dictionary. To the best of authors' knowledge, DTDL is the only DL model that understands the deep temporal structures of the data. Experiments on the Reference ED Data Set show an outstanding performance compared with the recent state-of-the-art algorithms in terms of precision, recall, accuracy, and F-score.**

***Index Terms*— Deep learning, dictionary learning (DL), energy disaggregation (ED), long short-term memory autoencoder (LSTM-AE).**

I. Introduction

ENERGY disaggregation (ED) also known as nonintrusive load monitoring is the problem of decomposing the whole electricity consumption signal of a residential, commercial, or industrial building into the signals of its appliances. The disaggregation algorithms can inform the service customers of their consumption patterns and recognize malfunctions in electricity appliances [1]. Furthermore, finding the detailed electricity consumption patterns of the customers helps energy suppliers to efficiently plan and operate power system networks [2].

Motivated by such beneficial applications, the energy society has been recently interested in finding accurate solutions to this problem. ED studies are generally categorized into two groups. The first group of approaches focuses on classifying electricity events rather than decomposing the energy consumption signals. Reference [3] is an early work in this area that leveraged transient and harmonic information with very high sampling rates; however, such data require costly hardware and monitoring devices. An event detection method is proposed in [4] based on the power ripple mitigation algorithm to recognize switching ON/OFF events of home appliances. Moreover, an enhanced version of the cumulative sum control chart algorithm is devised in [5] that models the switching events using predefined sliding windows that detect large variations in the electricity consumption data. Recent studies employed machine learning methods for supervised classification of electrical events [6]. A combination of artificial neural network (ANN) and particle swarm optimization (PSO) is presented in [7] to detect abrupt electricity consumption variations that reflect different switching ON/OFF events of home appliances. A feedforward ANN learns the switching patterns, while PSO optimizes its weights and biases. The graph signal processing (GSP) method [8] is a new class of event-based algorithms that detect the edges of electricity consumption signals. GSP models represent switching ON/OFF events by these edges in an unsupervised fashion. Leveraging the piecewise smoothness of power load signals, an unsupervised GSP algorithm is developed in [9] for edge detection in electricity signals. Moreover, a training-less GSP solution for online edge detection is proposed in [10]. In addition, in this line of research, a cloud-based online method is devised in [11] that clusters consumption patterns of various electrical appliances using the edge information obtained by the GSP.

The second group of algorithms aims to decompose the total electricity signal into its component devices. In this domain, hidden Markov model (HMM) [12] is a data-driven approach that casts ED to a Markov process problem to learn the state transition patterns of electricity signals. A novel combination of HMM and $L1$-norm edge detection is presented in [13] to extract the steady-state phase of electrical devices. Moreover, a hierarchical HMM is proposed in [14] to provide an accurate representation of home appliances with multiple built-in modes that correspond to distinct power consumption profiles. In addition, a hybrid load classification system is presented in [15] that combines HMMs with the K-nearest neighborhood (K-NN) clustering for power appliance disaggregation. The K-NN discriminates individual load patterns of different

Manuscript received September 1, 2018; revised May 13, 2019; accepted May 30, 2019. Date of publication July 10, 2019; date of current version May 1, 2020. This work was supported by the U.S. Department of Energy (DOE) Office of Electricity under Grant DE-OE 0000839. *(Corresponding author: Jianhui Wang.)*

M. Khodayar and J. Wang are with the Department of Electrical and Computer Engineering, Southern Methodist University, Dallas, TX 75275 USA (e-mail: mahdik@smu.edu; jianhui@smu.edu).

Z. Wang is with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011 USA (e-mail: wzy@iastate.edu).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2019.2921952

devices, while a set of HMMs learn to find state transitions between the load patterns.

Factorial HMM (FHMM) [16] is a novel variant of classic HMM that models the total energy signal as a random variable conditioned on multiple independent Markov chains that correspond to various electrical devices. An ED algorithm based on FHMM is presented in [17] that considers the current of appliances as load features and finds the mathematical relationship between the total current and the current of each device. Similar FHMM models are employed in [18] and [19] for industrial machinery power consumption monitoring. A bivariate version of HMM is presented in [20] that applies FHMMs to represent the joint active and reactive power of electricity consumption signals for various appliances.

Moreover, in the class of data-driven decomposition methods, Gillis and Morsi [21] proposed a cotraining semi-supervised approach that employs a set of wavelet features to represent energy signals and decompose them using nearest neighborhood classifiers. In addition, a support vector machine is defined in [22] to decompose energy signals using voltage–current (VC) trajectory as the signature for appliances. Similar VC formulation is used in [23] to disaggregate energy signals using transfer learning and convolutional neural networks. In this category of models, an extreme learning machine (ELM) is employed in [24] as a binary one-hot coding classifier that can categorize the electrical devices based on their consumption patterns. Furthermore, a hierarchical ELM is proposed in [25] to decompose electricity signals according to their extracted features obtained from an autoencoding neural network.

A. Related Work

In recent literature, dictionary learning (DL) and sparse modeling [26], [27] have shown significant achievements in various areas of signal processing [28]. DL-based ED algorithms formulate the total electricity signal as a sparse combination of individual energy snippets. Each snippet is a short-period time series that represents an operation mode of a device. A discriminative sparse coding algorithm is presented in [29] to decompose the total energy by a linear dictionary matrix whose columns (i.e., atoms) are the extracted energy snippets. Moreover, a novel supervised sparse model is devised in [30] to estimate a dictionary of snippets obtained from a multi-state finite-state machine. For each device, a dictionary learns its various operation modes (i.e., states) computed by the state machine corresponding to that device. In this category of models, an unsupervised DL algorithm is presented in [31] to minimize an $L1$-norm loss function that leads to robust dictionary atoms. The supervised version of robust DL [31] is proposed in [32] that models the sparse perturbations in the energy signals. In addition, an analysis co-sparse coding model is presented in [33] as a novel approach to DL that reduces the amount of required training data in ED problems. Furthermore, in this line of research, a nonparametric scalable DL is presented in [34] as a polynomial-time successive approximation. Using the submodularity per set-block, this method iteratively maximizes a set of global lower bounds on an ED objective function.

B. Paper Contributions

Recent ED research is dominated by DL-based algorithms. However, the existing DL models suffer from three crucial shortcomings that significantly impact their accuracy. This paper addresses the following shortcomings.

1) *Linearity Assumption:* Current DL algorithms seek to minimize a sparse linear reconstruction error corresponding to the data in the original ambient space. Each data point is modeled as a linear combination of the dictionary atoms that all lie on the same ambient space. However, in many real-world applications including ED, the relationships between different variables of the input data in the original ambient space are too complicated to be well-represented by linear mapping. Hence, in this paper, a more complex nonlinear feature extraction method is proposed to capture powerful features from the ambient space and significantly reduce the data dimensionality.
2) *Lack of Temporal Structure:* The existing DL algorithms are incapable of leveraging the temporal structures exist in the data. Therefore, they cannot provide reasonable accuracy for sequential datasets. However, in many applications including ED, the input data are time series in which the variables have strong temporal relationships. Deep learning for temporal feature extraction has shown promising performance in many areas of machine learning including renewable energy prediction [35]–[37], traffic forecasting [43], and load modeling [39]. In this paper, a novel recursive DL formulation is presented to address this challenge. The proposed recurrent formulation keeps track of the temporal changes in the data, hence learning the temporal manifold of the data.
3) *Atom-Feature Independence:* The existing DL algorithms assume that the input features fed to the DL model are independent of the dictionary atoms. Therefore, these features are not tuned to help better optimize the dictionary matrix in an informative way. Moreover, the dictionary does not provide instructive feedbacks to better optimize the input features. To address this concern, this paper presents a new end-to-end learning optimization to simultaneously train the dictionary and the input features, hence computing more discriminative features and dictionary atoms.

Motivated by the discussed drawbacks, this paper presents deep temporal DL (DTDL) as a novel deep learning algorithm for ED. The objective is to learn a complex nonlinear dictionary of energy signals that best describe the consumption patterns of electrical devices. In contrast to all existing DL models, DTDL finds an optimal dictionary of energy time series inside the latent space of a deep neural network rather than the original embedding space of the data. Hence, DTDL is not restricted by the linearity assumption. To learn the sequential structure of input energy signals, a long short-term

memory autoencoder (LSTM-AE) is presented as a recurrent deep neural network that learns powerful temporal states of the data. A novel optimization is proposed to simultaneously learn the dictionary and the corresponding sparse representation in the space of the LSTM-AE's latent features. Thus, DTDL can exploit valuable time-dependent features to improve the quality of its computed dictionary. The presented optimization alternates between refining the LSTM-AE's parameters and finding the optimal dictionary in an end-to-end fashion. Hence, the dictionary atoms and the LSTM-AE's temporal features pass informative knowledge to each other to find an optimal solution.

The rest of this paper is organized as the following. Section II defines the mathematical formulation of the ED problem. Moreover, Section III provides a conventional DL solution to this problem. In Section IV, the DTDL is proposed as a new deep learning algorithm to solve ED. The recurrent architecture as well as the mathematical optimization of the proposed method are discussed in this section. Section V explains a novel optimization algorithm for the DTDL that leads to a deep learning solution for ED. Section VI presents the disaggregation experiments on a real-world data set. The quantitative and qualitative comparisons with the state-of-the-art ED benchmarks are presented in this section. Finally, the conclusions of this research are stated in Section VII.

II. Problem Formulation

Let us assume there are L electric devices in a building and each device i consumes an energy signal $x_i(t)$ at each time $1 \leq t \leq T$. The aggregate consumption signal observed (recorded) by the smart meter is computed by

$$\overline{x}(t) = \sum_{i=1}^{L} x_i(t) \tag{1}$$

where $\overline{x}(t)$ is the total power consumed at time t. Observing the aggregated signals $\{\overline{x}(t)\}_{t=1}^{T}$, the goal is to recover the consumption signal of the individual appliances $1 \leq i \leq L$, i.e., the estimation of $\{x_i(t)\}_{t=1}^{T}$ for each valid i and t.

Let us consider an energy consumption data set C corresponding to a building that contains the energy signals of different devices through time (from $t = 1$ up to $t = T$). The consumption signals are broken into windows of length $\omega \ll T$ for all devices. For each device i, the consumption electricity in the time interval $[(k-1)\omega + 1, k\omega]$ is denoted by $y_i(k)$, called an energy snippet, for all $k = 1, 2, \ldots, K = (T/\omega)$. The corresponding aggregate signal is denoted by $\overline{y}(k)$. As a result, C is defined by $C = \{C_1, C_2, \ldots, C_K\}$ in which each data sample C_k is a tensor of the form $\langle y_1(k), y_2(k), \ldots, y_L(k), \bar{y}(k)\rangle$. The goal is to build a dictionary matrix $D \in \mathbb{R}^{\omega \times N}$ such that a solution of the following problem reveals the disaggregation of $\overline{y}(k)$:

$$\begin{aligned} \overline{y}(k) &= Da(k) = \sum_{j=1}^{N} a_j(k) D_{.,j} \\ D &= [D_1 D_2 \ldots D_L] \in \mathbb{R}^{\omega \times N} \quad D_i \in \mathbb{R}^{\omega \times N_i}. \end{aligned} \tag{2}$$

where D is a dictionary matrix of size $\mathbb{R}^{\omega \times N}$. Each column j of the dictionary, i.e., $D_{.,j}$, is a representative signal (also called an atom) for the device consumption signals $y_i(k)\, i \in [1, L]$, $k \in [1, K]$, that is, every signal $y_i(k)$ can be written as a linear combination of several columns (atoms) in D. $a(k)$ is a sparse coefficient vector that determines the coefficients of such a linear combination. Each element $a_j(k)$ decides the contribution of each column $D_{.,j}$ to the total consumption signal $\overline{y}(k)$.

III. Classic Dictionary Learning

In this section, the decomposition problem in (2) is cast to a classic DL (CDL) problem where the total signal $\overline{y}$ is decomposed by a dictionary matrix D and a sparse code a. As shown in (1), D can be decomposed into L subdictionaries $D_i \in \mathbb{R}^{\omega \times N_i}$, each corresponding to a device; hence, each signal $y_i(k)\, k \in [1, K]$ can be written as a linear combination of columns of the subdictionary D_i, while N_i is the number of these columns (atoms) defined for each device i. Therefore, the aggregate signal $\overline{y}(k)$ is a linear combination of the columns (atoms) in D_i each associated with a sparse coefficient vector $a^i(k)$ written by

$$\begin{aligned} \overline{y}(k) &= Da(k) \\ D &= [D_1 D_2 \ldots D_L] \in \mathbb{R}^{\omega \times N} \\ a(k) &= [a^1(k) a^2(k) \ldots a^L(k)] \in \mathbb{R}^N \quad a^i(k) \in \mathbb{R}^{N_i}. \end{aligned} \tag{3}$$

Since each device i has multiple consumption patterns corresponding to different operation modes, the objective is to extract useful consumption signatures (temporal patterns) through time to build subdictionaries D_i for each device i, as a matrix whose columns (atoms) can best represent the energy snippets $y_i(k)\, k \in [1, K]$. Moreover, the optimal sparse coefficients $a^i(k)$ need to be computed for all devices to reveal the contribution of each device in the total consumption signal $\overline{y}(k)$.

One can find the optimal sparse coefficients $a^*(k)$ for each k by solving the sparse coding problem with l_1 regularization as formulated by

$$a^*(k) = \text{argmin}_{a(k)} ||\overline{y}(k) - Da(k)||_2^2 + \lambda_1 ||a(k)||_1 \tag{4a}$$

$$\text{s.t. } 1^T a^i(k) \leq 1 a^i(k) \in \{0, 1\}^{N_i}. \tag{4b}$$

where $||\overline{y}(k) - Da(k)||_2^2$ is the signal reconstruction error, while $||a(k)||_1$ is the sparsity error with coefficient λ_1 that provides a tradeoff between the reconstruction accuracy and sparsity of the solution $a^*(k)$. The condition in (4b) makes sure that for each device i, only one column (atom) is found as its signature in $\overline{y}(k)$; thus, the device i with a nonzero element in $a^i(k)$ is operating/ON and its contribution to the total consumption $\overline{y}(k)$ is determined by

$$(D_i)_{.,j} (a^i(k))_j \quad \text{s.t. } (a^i(k))_j \neq 0 \tag{5}$$

where $(D_i)_{.,j}$ is the jth column of D_i, while $(a^i(k))_j$ is the jth entry of $a^i(k)$.

Furthermore, to find the optimal dictionary D with respect to the data set C with K data samples, one can minimize the following empirical cost function over the dictionary D and sparse coefficient matrix $A = [a(1) a(2) \ldots a(K)] \in \mathbb{R}^{N \times K}$:

$$\min_{D,A} \frac{1}{K} \sum_{k=1}^{K} (||\overline{y}(k) - Da(k)||_2^2 + \lambda_1 ||a(k)||_1) \tag{6a}$$

$$\text{s.t. } \|D_{.j}\|_2^2 \leq 1 \quad j = 1, 2, \ldots, N \tag{6b}$$

$$1^T a^i(k) \leq 1 a^i(k) \in \{0, 1\}^{N_i} \quad i = 1, 2, \ldots, L \quad k = 1, 2, \ldots, K \tag{6c}$$

Here, the constraint in (6b) prevents the dictionary from being arbitrarily large, since it can cause very small coefficient values in A, which makes the solution less informative.

IV. Deep Temporal Dictionary Learning: A New Paradigm Toward Sparse Coding

This section proposes the novel DTDL algorithm to solve the disaggregation problem formulated in (2). First, the mathematical shortcomings of CDL algorithms are discussed. Then, DTDL is presented as a novel paradigm toward DL and sparse coding to address these issues.

A. Mathematical Drawbacks of Classic DL

CDL optimization in (4a)–(6c) has three major shortcomings that motivate the need for a more powerful framework.

1) *Linearity of Solution:* CDL learns a linear D. As shown in Fig. 1, the optimization of the CDL model in (4a)–(6c) finds an approximation $\tilde{\overline{y}}(k)$ for the true value $\overline{y}(k)$ by finding dictionary atoms $D_{.,j}$ that are inside the original space of $y_i(k)\, i \in [1, L]\, k \in [1, K]$; however, if such a space is nonlinear (as in the case of most real-world applications including ED), the estimation value $\tilde{\overline{y}}(k)$ might not be in the original space S, making the estimation $Da(k)$ useless for modeling the true $\overline{y}(k)$. This motivates us to devise a novel nonlinear DL method based on deep learning to provide a nonlinear mapping from S to an appropriate transformed space S' in which $\tilde{\overline{y}}(k)$ can be well written as a combination of atoms of D as both $\overline{y}(k)$ and the columns $D_{.,j}$ lie on the same space S'. Learning such a nonlinear mapping, i.e., learning the transformed space S', is a crucial challenge solved by the DTDL.
2) *Lack of Sequential Structure:* The CDL cannot leverage the temporal patterns of sequential data sets; thus, the need for a recurrent optimization model that can capture useful temporal patterns from the underlying data, i.e., signals $y_i(k)$ inside the data set C, is raised. As the energy consumption signals in (2) are all time series, in this paper, a novel deep recurrent optimization model is proposed to address this issue.
3) *Dictionary-Feature Independence:* The CDL assumes a strong independence between the dictionary atoms $D_{.,j}$ and the input signal $\overline{y}(k)$; hence, training the dictionary using (6a)–(6c) does not lead to informative knowledge to update the representation of $\overline{y}(k)$. Moreover, learning the representation of the input signal $\overline{y}(k)$ does not yield useful knowledge to update the dictionary D. The proposed DTDL solves this issue by providing a novel optimization algorithm that alternates between updating input signal features and the dictionary.

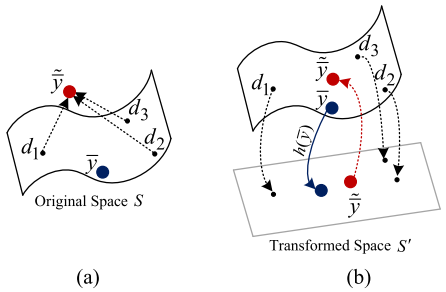


Fig. 1. (a) CDL: estimating the true consumption signal $\overline{y}(k)$ by dictionary atoms d_1, d_2, and d_3 inside the original nonlinear space S. (b) DTDL: transformation of S by a mapping function h to learn dictionary atoms inside the transformed space S'. The mapping provides a better estimation $\tilde{\overline{y}}(k)$ for $\overline{y}(k)$ when mapped back to S.

B. DTDL Recurrent Model

To tackle the presented challenges in Section IV-A, DTDL is proposed as a novel deep learning-based optimization for the disaggregation problem in (2). DTDL is a DL algorithm with a deep recurrent formulation to capture nonlinear sequential features that can help the model to better understand the behavior of the energy consumption temporal data, i.e., $y_i(k)$ signals. To learn each subdictionary D_i corresponding to each device i and the sparse code matrix A, DTDL learns N_i number of optimal $\omega-$ dimensional energy snippets $\hat{y}_i = \langle \hat{y}_i(1), \hat{y}_i(2), \ldots, \hat{y}_i(N_i) \rangle \in \mathbb{R}^{\omega \times N_i}$ for each device i, such that the elements of $\hat{y}_i$ best represent the energy snippets $y_i(k)$ for $k \in [1, K]$. In other words, for each device i, every $y_i(k)$ can be written as a linear combination of elements of $\hat{y}_i$; hence, one can conclude that the optimal subdictionary is found by $D_i = \hat{y}_i \in \mathbb{R}^{\omega \times N_i}$.

Assuming that the energy snippets $y_i(k)\, k \in [1, K]$ lie on a nonlinear manifold M, to find $D_i = \hat{y}_i$, DTDL learns a nonlinear transformation $F_{\text{enc}} : \mathbb{R}^{\omega} \to \mathbb{R}^{d}$ that encodes each energy snippet $y_i(k) \in \mathbb{R}^{\omega}$ by a d-dimensional latent feature vector $h(y_i(k)) \in \mathbb{R}^{d}$ that captures the fundamental nonlinear temporal relationships of the variables in energy snippet $y_i(k)$. The latent feature vector is further decoded by a nonlinear mapping $F_{\text{dec}} : \mathbb{R}^{d} \to \mathbb{R}^{\omega}$ that maps the extracted $h(y_i(k))$ in the nonlinear (transformed) space to the observed $y_i(k)$ in the original space, hence learning a powerful nonlinear embedding function h that is capable of reconstructing the original consumption signal. Such nonlinear mapping h is implemented by F_{enc} and mapped back to the original space of energy snippets by F_{dec}. While $h(y_i(k))$ is computed (i.e., the optimal F_{enc} and F_{dec} are found) for all $y_i(k)$ in the data set C, DTDL learns $D_i = \hat{y}_i$ inside the transformed space corresponding to h, hence learning the nonlinear dictionary $D_i = \hat{y}_i$ for the energy snippets $y_i(k)\, k \in [1, K]$ for each device i.

Since DTDL is working with the temporal data $y_i(k)$, an LSTM-AE neural network is devised using a deep learning-based recurrent formulation to capture the sequential structure of the data. The presented recurrent model can capture the sequential relationships of variables in $y_i(k)$ by learning the temporal manifold of $y_i(k)$. As shown in Fig. 2, the proposed LSTM-AE is an LSTM network with 2ω temporal

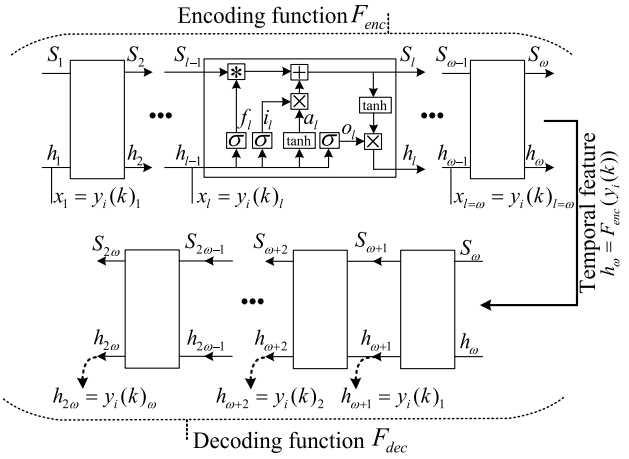


Fig. 2. Structure of the proposed LSTM-AE.

states S_i $i = 1, 2, \ldots, 2\omega$. The first ω states serve to model F_{enc} that maps $y_i(k)$ to $h(y_i(k))$ in ω iterations. At each iteration $1 \leq l \leq \omega$, an input element $y_i(k)_l \in \mathbb{R}$ is observed by the LSTM unit and the temporal state S_{l-1} is updated to S_l using the following recurrent formulation:

$$\begin{aligned} x_l &= y_i(k)_l \\ a_l &= \tanh(W_a x_l + U_a h_{l-1} + b_a) \\ i_l &= \text{Sigm}(W_i x_l + U_i h_{l-1} + b_i) \\ f_l &= \text{Sigm}(W_f x_l + U_f h_{l-1} + b_f) \\ o_l &= \text{Sigm}(W_o x_l + U_o h_{l-1} + b_o) \\ S_l &= f_l \circ S_{l-1} + i_l \circ a_l \\ h_l &= o_l \circ \tanh(S_l) \end{aligned} \tag{7}$$

where x_l is the input vector, while i_l, f_l, and o_l are the m-dimensional input gate, forget gate, and output gate decision variables at iteration l, respectively. a_l is the input activation with bias b_a, and h_l is the output vector of the LSTM at iteration l, which contains the deep temporal features obtained from the input sequence from iteration 1 up to l. The parameters b_i, b_f, b_o, and b_a are the m-dimensional bias vectors, while W_i, W_f, W_o, and W_a are weight parameters inside $\mathbb{R}^m$. Moreover, U_i, U_f, U_o, and U_a are the weight matrices in $\mathbb{R}^{m\times m}$. All bias and weight parameters are tunable parameters that are learned to find the optimal temporal state S_l as well as the optimal temporal feature h_l at each iteration l. When $l = \omega$, the whole $y_i(k)$ signal has been observed and $h_l = h_\omega = h(y_i(k))$ is obtained by the LSTM as the temporal feature vector of the whole consumption signal $y_i(k)$; thus, the first ω iterations of the LSTM implement F_{enc} mapping each energy snippet $y_i(k)$ to the corresponding embedding vector h.

As shown in Fig. 2, the iterations $\omega + 1 \leq l \leq 2\omega$ reconstruct $y_i(k)$. At each time instance l, an output feature $h_l = y_i(k)_{l-\omega}$ is generated by the LSTM to model F_{dec} that maps the resulting temporal features of F_{enc}, i.e., $h(y_i(k))$, to the original consumption snippet $y_i(k)$. This leads to learning the nonlinear temporal manifold of the energy snippets $y_i(k)\, i = 1, 2, \ldots, L\ k = 1, 2, \ldots, K$ as the LSTM-AE learns sequential features $h_\omega = h(y_i(k))$ in its iteration $l = \omega$ that are so powerful that can reconstruct the original energy snippets $y_i(k)$.

C. DTDL Optimization Program

DTDL learns the subdictionaries D_i and the sparse coefficient matrix A in the latent space of $h_\omega = h(y_i(k))$, that is, when $y_i(k)$ is mapped in the ωth iteration of the LSTM to the feature vector h_ω, the columns (atoms) of the corresponding subdictionary D_i are learned to represent h_ω so that the linear combinations of the atoms (columns) of each D_i would be able to yield the feature vectors $h_\omega = h(y_i(k))$ for all $k = 1, 2, \ldots, K$. Such linear combinations are determined by the sparse code matrix A.

DTDL defines the following novel optimization program for the problem of learning the optimal D and A, while finding the optimal mappings F_{enc} and F_{dec}:

$$\begin{aligned} &\min_{F_{\text{enc}}, F_{\text{dec}}, \{D_i\}_{i=1}^L} \quad J = J_1 + \lambda_2 J_2 + \lambda_3 J_3 + \lambda_4 J_4 \\ &J_1 = \frac{1}{L}\sum_{i=1}^{L}\frac{1}{K}\sum_{k=1}^{K}\left(||F_{\text{enc}}(y_i(k)) - D_i a^i(k)||_2^2 + \lambda_1 ||a^i(k)||_1\right) \\ &J_2 = \sum_{j=1, j\neq i}^{L} \left|\left|D_i^T D_j\right|\right|_F^2 \\ &J_3 = \frac{1}{L}\sum_{i=1}^{L}\frac{1}{K}\sum_{k=1}^{K}\left(||F_{\text{dec}}(F_{\text{enc}}(y_i(k))) - y_i(k)||_2^2\right) \\ &J_4 = \left(||W_f||_F^2 + ||W_i||_F^2 + ||W_o||_F^2 + ||U_f||_F^2 + ||U_i||_F^2 + ||U_o||_F^2 + ||b_f||_2^2 + ||b_i||_2^2 + ||b_o||_2^2\right) \\ &\text{s.t.} \ \frac{1}{K}\sum_{k=1}^{K}|1^T a^i(k) - 1^T a^i(k+1)| = 0 \quad \forall\ i \\ &\left\|(D_i)_{.,j}\right\|_2^2 \leq 1 \quad i = 1, 2, \ldots, L \ \ j = 1, 2, \ldots, N_i. \end{aligned} \tag{8}$$

Here, J_1 is the data reconstruction cost function to compute the difference between $h_{l=\omega} = h(y_i(k)) = F_{\text{enc}}(y_i(k))$ and the linear combination of the atoms in the subdictionary D_i computed by $D_i a^i(k)$. Such difference is computed for all devices i and all time intervals k in the data set C. The term $\lambda_1||a^i(k)||_1$ ensures sparsity for the solution of A. J_2 is the cross-dictionary incoherence term; minimizing this error promotes incoherence between two subdictionaries D_i and $D_{j\neq i}$, that is, this error term is in favor of having distinct subdictionary atoms for different devices. J_3 is the LSTM-AE's reconstruction error term, which is the distance between the output of LSTM-AE generated at iterations $\omega+1 \leq l \leq 2\omega$, i.e., $F_{\text{dec}}(F_{\text{enc}}(y_i(k)))$, and the desired output $y_i(k)$ for all devices i and all time intervals k. J_4 is the regularization error defined to control the magnitude of the LSTM-AE's parameters. Large parameters might lead to the overfitting problem; thus, J_4 is defined over LSTM parameters of (7) to avoid this issue. The first constraint in (8) satisfies the temporal smoothness prior. Note that, for any device i, the term $|1^T a^i(k) - 1^T a^i(k+1)|$ is zero except at intervals when it turns ON/OFF. Given the fact that such switching happens in very small periods of time compared

to the whole time period, the term $(1/K)\sum_{k=1}^{K}|1^T a^i(k) - 1^T a^i(k+1)|$ is minimized for all devices. Finally, the constraint $\|(D_i)_{.,j}\|_2^2 \leq 1$ is assumed to avoid each subdictionary from obtaining arbitrary large entries as it would cause very small entries in the coefficient matrix A that can lead to trivial solutions.

V. DTDL Optimization Solution

The optimization program in (8) is not jointly convex with respect to $\{F_{\text{enc}}, F_{\text{dec}}, D_i\}_{i=1}^{L}$, and A. As a result, an iterative algorithm is proposed that alternates between optimization of the functions F_{enc} and F_{dec}, and finding the optimal variables $\{D_i\}_{i=1}^{L}$ and A. The proposed algorithm addresses three subproblems alternately as explained in the following.

A. Optimization of the LSTM-AE Mappings F_{enc} and F_{dec}

Having a fixed D and A, to optimize F_{enc} and F_{dec} in the objective (8), one needs to solve the following optimization:

$$\begin{aligned}\min_{F_{\text{enc}},F_{\text{dec}}}\ & \tilde{J} \\ =& \left\{\frac{1}{L}\sum_{i=1}^{L}\frac{1}{K}\sum_{k=1}^{K}\Big[\big(||F_{\text{enc}}(y_i(k)) - D_i a^i(k)||_2^2\big)\right. \\ &\left. + \big(||F_{\text{dec}}(F_{\text{enc}}(y_i(k))) - y_i(k)||_2^2\big)\Big]\right\} \\ &+ \lambda_4\big(||W_f||_F^2 + ||W_i||_F^2 + ||W_o||_F^2 + ||U_f||_F^2 + ||U_i||_F^2 \\ &+ ||U_o||_F^2 + ||b_f||_2^2 + ||b_i||_2^2 + ||b_o||_2^2\big). \end{aligned} \tag{9}$$

As explained in Section IV-C, the term $F_{\text{enc}}(y_i(k))$ in (9) is the output of the ω-th iteration of the LSTM-AE denoted by $h_{l=\omega}$; hence, DTDL needs to train the LSTM unit to output $h_{l=\omega} = D_i a^i(k)$ at this iteration. Moreover, $F_{\text{dec}}(F_{\text{enc}}(y_i(k)))$ is the output of the LSTM-AE in iterations $\omega+1$ through 2ω, i.e., $F_{\text{dec}}(F_{\text{enc}}(y_i(k))) = [h_{\omega+1}h_{\omega+2}\ldots h_{2\omega}]$. Therefore, DTDL needs to train LSTM-AE to satisfy $[h_{\omega+1}h_{\omega+2}\ldots h_{2\omega}] = y_i(k)$ in (9). Let us define the following notations for the LSTM parameters at each iteration l:

$$\text{gates}_l = \begin{bmatrix} a_l \\ i_l \\ f_l \\ o_l \end{bmatrix},\ W = \begin{bmatrix} W_a \\ W_i \\ W_f \\ W_o \end{bmatrix},\ U = \begin{bmatrix} U_a \\ U_i \\ U_f \\ U_o \end{bmatrix},\ b = \begin{bmatrix} b_a \\ b_i \\ b_f \\ b_o \end{bmatrix}. \tag{10}$$

To minimize $\tilde{J}$ in (9) after observing each $y_i(k)$, we compute the gradient of $\tilde{J}$ with respect to the LSTM's output h_l using

$$\begin{aligned}\Delta_l &= \frac{\partial \tilde{J}}{\partial h_l}(y_i(k)) \\ &= \begin{cases} 2[F_{\text{enc}}(y_i(k)) - D_i a^i(k)] & l = \omega \\ 2[[h_{\omega+1}h_{\omega+2}\ldots h_{2\omega}] - y_i(k)] & l \geq \omega+1 \end{cases}\end{aligned} \tag{11}$$

Thus, for each LSTM iteration $l = 1, 2, \ldots, \omega$, DTDL computes the partial derivatives of $\tilde{J}$ with respect to various LSTM's gates by

$$\begin{aligned}\delta h_l &= \Delta_l + \Delta h_l \\ \delta S_l &= \delta h_l \circ o_l \circ (1 - \tanh^2(S_l)) + \delta S_{l+1} \circ f_{l+1} \\ \delta a_l &= \delta S_l \circ i_l \circ (1 - a_l^2) \\ \delta i_l &= \delta S_l \circ a_l \circ i_l \circ (1 - i_l) \\ \delta f_l &= \delta S_l \circ S_{l-1} \circ f_l \circ (1 - f_l) \\ \delta o_l &= \delta h_l \circ \tanh(S_l) \circ o_l \circ (1 - o_l) \\ \delta x_l &= W^T.\delta\text{gates}_l \\ \Delta h_{l-1} &= U^T.\delta\text{gates}_l \end{aligned} \tag{12}$$

where $\circ$ is the Hadamard product. Considering (9)–(12), the partial derivatives of $\tilde{J}$ with respect to the LSTM's parameters W, U, and b are computed by

$$\begin{aligned}\delta W &= \sum_{l=0}^{2\omega}\delta\text{gates}_l \otimes x_l + \lambda_4 W \\ \delta U &= \sum_{l=0}^{2\omega-1}\delta\text{gates}_{l+1} \otimes h_l + \lambda_4 U \\ \delta b &= \sum_{l=0}^{2\omega}\delta\text{gates}_{l+1} + \lambda_4 b. \end{aligned} \tag{13}$$

Having (12), the LSTM-AE model (which is an implementation of F_{enc} and F_{dec}) is updated using the following update rule based on the gradient descent method:

$$\begin{aligned} W^{\text{new}} &\leftarrow W - \eta\delta W \\ U^{\text{new}} &\leftarrow U - \eta\delta U \\ b^{\text{new}} &\leftarrow b - \eta\delta b. \end{aligned} \tag{14}$$

Here, W^{new}, U^{new}, and b^{new} are, respectively, the updated parameters for W, U, and b using the gradient descent update rule (13) after observing each $y_i(k)\, i = 1, 2, \ldots, L\ \ k = 1, 2, \ldots, K$ in 2ω iterations. η is the learning rate that determines how strong each update can be.

B. Optimization of the Dictionary D

Given the fixed mappings F_{enc} and F_{dec}, as well as some fixed sparse code matrix A, the main optimization in (8) would have the following form by which DTDL seeks to optimize D:

$$\begin{aligned}\min_{D=[D_1\ D_2\ldots D_L]}\ & \bar{J} = \frac{1}{L}\sum_{i=1}^{L}\frac{1}{K}\sum_{k=1}^{K}\big(||F_{\text{enc}}(y_i(k)) - D_i a^i(k)||_2^2\big) \\ &+ \lambda_2\sum_{j=1, j\neq i}^{L}\big||D_i^T D_j\big||_F^2 \\ \text{s.t. }& \|(D_i)_{.,j}\|_2^2 \leq 1 \quad i = 1, 2, \ldots, L \quad j = 1, 2, \ldots, N_i. \end{aligned} \tag{15}$$

Here, the cross subdictionary incoherence error term $\lambda_2\sum_{j=1, j\neq i}^{L}||D_i^T D_j||_F^2$ tries to enforce the resulting subdictionaries of different devices $i \neq j$ to have distinct dictionary atoms. To investigate the effect of such an error term on the accuracy of the solution, two different settings are assumed to solve (15). The first case assumes a zero coefficient λ_2, while the second case considers a nonzero λ_2. The solutions under both of these assumptions are investigated in the following.

1) No Subdictionary Incoherence Error ($\lambda_2 = 0$) in (15): In this setting, there is no incoherence error; hence, each two devices might contain similar atoms in their corresponding subdictionaries. This changes the optimization of (15) to a least squares problem with quadratic constraints; thus, DTDL solves (15) using Lagrangian multipliers. First, let us define the Lagrangian in the following form using the Lagrangian multipliers $\phi = [\phi_{i,j} \geq 0] \quad i = 1, 2, \ldots, L \;\; j = 1, 2, \ldots, N_i$:

$$\mathcal{L}(D, \phi) = \frac{1}{L}\sum_{i=1}^{L}\frac{1}{K}\sum_{k=1}^{K}\left(||F_{\text{enc}}(y_i(k)) - D_i a^i(k)||_2^2\right) + \sum_{i=1}^{L}\sum_{j=1}^{N_i}\phi_{i,j}\left(\|(D_i)_{.,j}\|_2^2 - 1\right) \quad (16)$$

Considering a set of multipliers as $\tilde{\phi} = [\tilde{\phi}_j \geq 0]_{j=1}^{N}$, one can rewrite (16) using

$$\mathcal{L}(D, \phi) = \frac{1}{L}\sum_{i=1}^{L}\frac{1}{K}\sum_{k=1}^{K}\left(||F_{\text{enc}}(y_i(k)) - D_i a^i(k)||_2^2\right) + \sum_{j=1}^{N}\tilde{\phi}_j\left(\|D_{.,j}\|_2^2 - 1\right). \quad (17)$$

Having $(\partial L(D, \phi)/\partial D) = 0$, the following analytical solution is computed as an optimal solution for (17):

$$D = F_{\text{enc}}^{\text{Total}} A^T (AA^T + \Sigma)^{-1}$$
$$F_{\text{enc}}^{\text{Total}} = \left\langle \begin{matrix} F_{\text{enc}}(y_{i=1}(1)) \ldots F_{\text{enc}}(y_{i=1}(K)) \\ \ldots \\ F_{\text{enc}}(y_{i=L}(1)) \ldots F_{\text{enc}}(y_{i=L}(K)) \end{matrix} \right\rangle \in \mathbb{R}^{\text{d}\times(L*K)} \quad (18)$$

where $F_{\text{enc}}^{\text{Total}} \in \mathbb{R}^{\text{d}\times(L*K)}$ is a vector of all $F_{\text{enc}}(y_{i=l}(k))$ for all $i = 1, 2, \ldots, L$ and $k = 1, 2, \ldots, K$. Note that $d = \dim(h_{l=\omega})$ is the dimension of the temporal feature vector $h_{l=\omega} = F_{\text{enc}}(y_i(k))$; also, Σ is computed by $\Sigma = (L * K)\text{diag}(\phi) \in \mathbb{R}^{\text{N}\times N}$. Therefore, the corresponding Lagrangian dual function is written as

$$\begin{aligned}\mathcal{L}_{\text{dual}}(\phi) &= \mathcal{L}(D, \phi) \\ &= \frac{1}{L}\sum_{i=1}^{L}\frac{1}{K}\sum_{k=1}^{K}\left\| F_{\text{enc}}(y_i(k)) - F_{\text{enc}}^{\text{Total}} A^T (AA^T + \Sigma)^{-1} a^i(k)\right\|_2^2 \\ &\quad + \sum_{j=1}^{N}\tilde{\phi}_j\left(\left\| F_{\text{enc}}^{\text{Total}} A^T (AA^T + \Sigma)^{-1} u_i\right\|_2^2 - 1\right) \end{aligned} \quad (19)$$

with the ith unit vector denoted by $u_i \in \mathbb{R}^N$. Leveraging the gradient descent method, the dual Lagrangian $\mathcal{L}_{\text{dual}}(\phi)$ in (19) is maximized with respect to $\tilde{\phi} = [\tilde{\phi}_j \geq 0]_{j=1}^{N}$. The gradient of the dual for any $\tilde{\phi}_j$ is computed by

$$\frac{\partial \mathcal{L}_{\text{dual}}(\phi)}{\partial \tilde{\phi}_j} = \left\| F_{\text{enc}}^{\text{Total}} A^T (AA^T + \Sigma)^{-1} u_i \right\|_2^2 - 1. \quad (20)$$

When the optimal $\tilde{\phi}$ is computed using gradient descent, the optimal dictionary D is estimated by (18) using the optimal $\Sigma = (L * K)\text{diag}(\phi)$.

2) Nonzero Subdictionary Incoherence Error ($\lambda_2 \neq 0$) in (15): When the subdictionary incoherence error term is considered, i.e., $\lambda_2 \neq 0$, one needs to optimize (15). Applying gradient descent, the error $\bar{J}$ in (15) is minimized with respect to each subdictionary for each training data $y_i(k)$ using the following gradient value for each subdictionary D_i:

$$\begin{aligned}\frac{\partial \overline{J}}{\partial D_i}(y_i(k)) &= D_i a^i(k)(a^i(k))^T - F_{\text{enc},i}(a^i(k))^T \\ &\quad + \lambda_2 \sum_{j=1, j\neq i}^{L}\left(D_i^T D_j\right) D_i \\ F_{\text{enc},i} &= \langle F_{\text{enc}}(y_i(1)) \ldots F_{\text{enc}}(y_i(K))\rangle \in \mathbb{R}^{d\times K}. \end{aligned} \quad (21)$$

Using (21), one can minimize the optimization error $\bar{J}$ with respect to every subdictionary D_i, hence optimizing the whole dictionary D.

C. Optimization of the Sparse Code Matrix A

When F_{enc}, F_{dec}, and D are fixed, one can optimize the coefficient matrix A while observing each signal $y_i(k)$. Let us write the main optimization program in (8) in the following form where the main objective J in (8) is optimized with respect to each $a^i(k)$ in A:

$$\begin{aligned}&\min_{a^i(k)} \overline{\overline{J}} = ||F_{\text{enc}}(y_i(k)) - D_i a^i(k)||_2^2 + \lambda_1 ||a^i(k)||_1 \\ &\text{s.t. } |1^T a^i(k) - 1^T a^i(k+1)| = 0 \quad \forall i, k. \end{aligned} \quad (22)$$

Here, to satisfy the constraint $|1^T a^i(k) - 1^T a^i(k+1)| = 0$ for all i and k, this condition is rewritten using a binary matrix G

$$\begin{aligned}\text{AG} &= 0A \in \mathbb{R}^{N\times(K*L)} G \in \mathbb{R}^{(K*L)\times(K*L)} \\ A &= (a^{i=1}(1) a^{i=2}(1) \ldots a^{i=L}(1) \\ &\quad \ldots a^{i=1}(K) a^{i=2}(K a^{i=L}(K)) \\ &= (A_{.,1} A_{.,2} \ldots A_{.,j} \ldots A_{.,(K*L)}) j = (k-1)L + i \\ G_{i,j} &= \begin{cases} 1 & i = j \\ -1 & i = j + L \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \quad (23)$$

Having the constraint $AG = 0$, i.e., $G^T A^T = 0$, one can rewrite optimization (22) to solve for each column $a^i(k) = A_{.,j}\, j = (k-1)L + i$ for all energy snippets $y_i(k)$ by

$$\begin{aligned}&\min_{A,\, j=(k-1)L+i} \sum_{i=1}^{L}\sum_{k=1}^{K} ||F_{\text{enc}}(y_i(k)) - DA_{.,j}||_2^2 + \lambda_1 ||A_{.,j}||_1 \\ &\text{s.t. } \sum_{i'=1}^{K*L}\sum_{j'=1}^{K*L} G_{j',i'} A_{.,j} = 0. \end{aligned} \quad (24)$$

Solving (24) by the proximal Jacobian alternating direction method of multipliers (ADMM) [40], the optimal sparse coefficient matrix is computed. Note that the dimension of G does not add much computational burden to the optimization program as a large portion of G's entries are zero.

D. DTDL Disaggregation Solution

Algorithm 1 shows the structure of the proposed ED algorithm that solves the optimization problem (2) using the presented DTDL model. Here, F_{enc}, F_{dec}, $\{D_i\}_{i=1}^{L}$, and A are optimized using an iterative algorithm alternating among the optimizations (9), (15), and (24). The optimizations are executed repeatedly until the average change in the dictionary entries is less than a small threshold $\varepsilon > 0$.

Algorithm 1 DTDL-Based Disaggregation Algorithm

Inputs: Energy signals of all devices $y_i(k) k \in [1, K] i \in [1, L]$

Outputs: Optimal Dictionary $D = D^*$ and sparse coefficient $a = a^*$ for the disaggregation problem $Y = Da$ where Y is a test aggregate consumption signal

1: **Repeat:**

2: Optimize (9) to update F_{enc}, F_{dec}

3: Optimize (15) to update the dictionary D

4: Optimize (24) to update the sparse coefficients A

5: **Until** Convergence (changes in dictionary entries are less than $\varepsilon > 0$)

6: Test the model: Given the optimal dictionary D^*, compute optimal coefficient vector a^* for an aggregate energy consumption signal Y :

$$a^* = \min_a \left\| F_{enc}(Y) - D^* a \right\|_2^2 + \lambda_1 ||a||_1 \tag{25}$$

In the test time, the optimal dictionary D^* is used in (25) to obtain the optimal coefficients a^* for some test aggregate energy signal Y of a building. Having the optimal dictionary D^*, the optimal coefficients a^* show the contribution of each device in the total electricity consumption Y. One can simply compute such contributions using (5).

VI. Simulation Results

A. Data Set

The proposed DTDL disaggregation algorithm is evaluated on the real-world Reference ED Data Set (REDD) [12], a large publicly available data set for electricity disaggregation problems. The data set contains power consumption signals of five houses with around 20 different appliances at each house. The electricity signals of each device as well as the total consumption are available for two weeks with a high frequency sampling rate $f = 15$ kHz.

Knowing that low-frequency sampling leads to a more practical energy measurement that is less costly and more challenging for ED, DTDL is trained and evaluated on low-frequency data. In this paper, a sampling rate $f = 1$ Hz is applied to collect the energy signals. DTDL is trained and validated using the data corresponding to the first week; 80% of these samples are used to train and the rest are used to validate the model to find the optimal hyperparameters. The samples of the second week are applied to test the model.

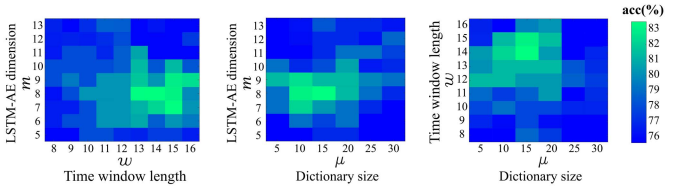


Fig. 3. Validation accuracy of DTDL with different configurations of LSTM-AE dimension, window length, and dictionary size.

B. Disaggregation Accuracy Metrics

Let us assume the test aggregate consumption signal Y contains K time intervals (windows) of length ω, each denoted by $Y(k)\, k = 1, 2, \ldots, K$. Signal $Y(k)$ is the summation of energy signals (energy snippets) $Y_i(k)\, i = 1, 2, \ldots, L$, that is, each device $1 \le i \le L$ consumes $Y_i(k)$ at the time interval k. In addition, let us denote the estimation of $Y_i(k)$ by $\hat{Y}_i(k) = D_i a^i(k)$ obtained by Algorithm 1. The disaggregation accuracy is computed by

$$\text{acc} = \left(1 - \frac{\sum_{k=1}^{K} \sum_{i=1}^{L} \|\hat{Y}_i(k) - Y_i(k)\|_1}{2 \sum_{k=1}^{K} \|Y(k)\|_1}\right) \times 100\%. \tag{26}$$

Here, the factor 2 in the denominator is due to the fact that the absolute value leads to double counting errors.

To have a comprehensive comparison, the precision, recall, and the F-score are computed at the device level. At each time period k, a binary "ON/OFF" value indicates whether each device i is operating ($a^i(k)$ is nonzero) or not ($a^i(k)$ is zero). Precision P determines what portion of the estimated ON/OFF decisions for a device truly belongs to that device, while recall R measures what portion of the ON/OFF value for one device is correctly estimated. F-score is the harmonic mean of P and R that combines these two metrics by

$$F_{\text{score}} = \frac{2 \times P \times R}{P + R}. \tag{27}$$

C. Experimental Settings and Model Validation

In this paper, the learning rate η of the LSTM-AE's update rule (14) is set to be 0.01 and the dictionary convergence threshold ε in Algorithm 1 is set to be 0.05. DTDL has several hyperparameters including the LSTM-AE's latent feature dimension m, the time window length ω, and the number of dictionary atoms μ considered for each device. To find an optimal configuration of hyperparameters, a validation search space is defined that contains all different configurations of $5 \le m \le 13$ and $8 \le \omega \le 16$ with $\mu \in \{5, 10, 15, 20, 25, 30\}$. For each configuration in this search space, DTDL is trained on the training set and evaluated on the validation set to compute the corresponding validation accuracy acc defined in (26). The configuration with the highest validation accuracy acc is chosen as the optimal model that is further evaluated on the testing set.

Fig. 3 shows the validation acc of DTDL averaged over all houses in the data set. As shown in this plot, the optimal configuration has $m = 8$ with a disaggregation accuracy of 83.71%. Increasing m to larger values would grow the

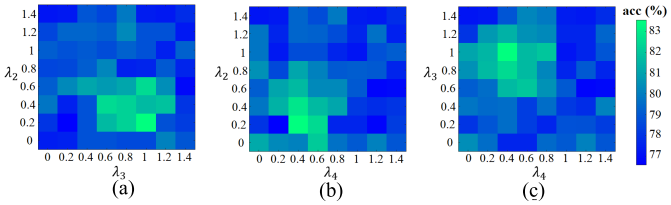


Fig. 4. Validation accuracy of DTDL with different configurations of error coefficients λ_2, λ_3, and λ_4. The contribution of various error terms to the accuracy of the disaggregation model is shown in terms of the accuracy metric in (26). (a) $\lambda_4 = 0.6$. (b) $\lambda_3 = 1.2$. (c) $\lambda_2 = 0.4$.

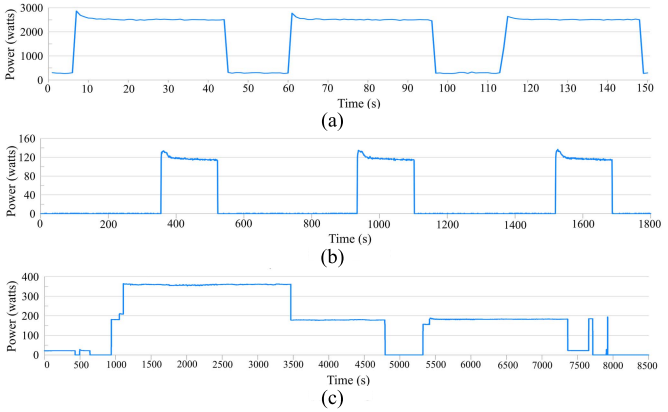


Fig. 5. Consumption pattern of (a) washer/dryer, (b) refrigerator, and (c) lighting device during their operation time for House 3 in the REDD data set.

generalization capability (i.e., nonlinear capacity) of the LSTM unit; however, it would also make the LSTM prone to overfitting; therefore, the moderate value of $m = 8$ is the optimal choice. It is also shown that the window size $\omega = 14$ leads to the highest validation accuracy. Note that smaller windows would degrade the accuracy as the transients would be overemphasized when learning D. Moreover, larger time windows would lead to observing different dynamics/operation modes in just a single time window, hence decreasing the disaggregation accuracy. As shown in Fig. 3, the optimal model with the highest validation accuracy contains $\mu = 15$ dictionary atoms per device. Having larger number of atoms would increase the likelihood of overfitting that declines the validation accuracy. In addition, having smaller number of atoms would cause the model to miss useful energy patterns that contain important operation modes of various devices in the data set.

To analyze the contribution of different error terms J_1,J_2,J_3, and J_4 defined in optimization (8) to the quality and accuracy of our ED solution, the validation accuracy is computed using different combinations of error coefficients λ_2, λ_3, and λ_4. Fig. 4 shows the validation acc averaged over all houses for such configurations of the objective function J. In this plot, each error coefficient is chosen from the set $\Lambda = \{0, 0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4\}$. As shown in Fig. 4, the optimal configuration is $\langle \lambda_2, \lambda_3, \lambda_4 \rangle = \langle 0.2, 1.0, 0.4 \rangle$.

As discussed in Section IV-C where the DTDL's optimization is proposed, J_2 is the cross dictionary incoherence error that enforces the devices to have dissimilar consumption patterns. The optimal coefficient $\lambda_2 = 0.2$ shows that this error term should have a relatively low (but more than zero) value, which means that, for some devices, the assumption of having dissimilar energy snippets is true (e.g., refrigerator and lighting devices), while for other devices, the energy snippets might have similar behaviors. For instance, as shown in Fig. 5, the devices with rotary components (i.e., motors) such as refrigerator and washer/dryer have similar consumption patterns that are different from lighting appliances.

J_3 is the reconstruction error term that makes sure that the LSTM-AE has learned useful temporal features that are strong enough to reconstruct the original consumption signals. The optimal coefficient of this error term is $J_3 = 1.0$, which is relatively high. This shows that learning powerful temporal features help the model to find more accurate disaggregation solutions. Therefore, the high value of J_3 justifies the DTDL for learning the transformed space S' using the presented LSTM-based architecture. Moreover, the regularization coefficient J_4 has a moderate value of 0.4 that shows the LSTM-AE can successfully avoid the overfitting problem by restricting the magnitude of its parameters.

D. Benchmarks

The proposed DTDL disaggregation model is compared with various recent ED benchmarks including simple mean prediction (SMP) [41], FHMM [19], [20], approximate MAP inference (AMAPI) [14], [42], hierarchical FHMM (HieFHMM) [15], and powerlet-based ED (PED) [7], [29]. Moreover, the CDL algorithm presented in Section III is considered as a baseline to better show the merit of deep learning in the area of sparse coding and DL. The compared benchmarks are briefly discussed in the following.

1) Simple Mean Prediction: SMP is recently introduced in [41] as a fundamental baseline for ED. In the training stage, this algorithm observes the total energy signal of a home as well as the energy signal of its individual devices. For each device, a consumption percentage is computed as the ratio of the energy consumed by that device to the total energy consumption. Then, during the test time, the total energy signal is disaggregated according to this ratio at all time steps.

2) Factorial HMM: FHMM is a recent signal disaggregation model presented in [20]. This benchmark is an HMM with a discrete hidden state $x_t^{(i)}$ for each device i at each time instance t. Given $x_t^{(i)}$, at each time step t, the device i is assumed to consume the real-valued energy $y_t^{(i)}$. At the training time, the HMM observes energy signals of the whole training set to compute a Gaussian posterior probability density function $P(y_t^{(i)})$ for each consumption signal $y_t^{(i)}$. In this paper, following [19] and [20], four HMM states are defined for each device while 20 devices are assumed for each home; hence, the FHMM learns $4^{20} \cong 10^{12}$ different hidden states to compute all $y_t^{(i)}$ variables in the REDD data set. The Baum–Welch expectation maximization algorithm is employed to train the HMM to estimate $P(y_t^{(i)})$ for all devices i at all time instances t. Moreover, the blocked Gibbs

TABLE I
DISAGGREGATION ACCURACY OF VARIOUS BENCHMARKS

	House					
Methods	**1**	**2**	**3**	**4**	**5**	**Average**
SMP	41.4	39.0	46.7	52.7	33.7	42.7
FHMM	71.5	59.6	59.6	69.0	62.9	64.5
AMAPI	73.2	61.4	62.3	60.1	66.3	64.7
HieFHMM	75.6	73.4	60.5	51.2	71.0	66.3
CDL	74.0	61.8	60.0	59.2	78.4	66.7
PED	81.6	79.0	61.8	58.5	79.1	72.0
DTDL	**83.5**	**84.7**	**64.6**	**74.3**	**85.6**	**78.5**

sampling method [43] is used to compute $y_t^{(i)}$ from the HMM's probability distribution $P(y_t^{(i)})$ for all time steps t.

3) Approximate MAP Inference: This model is recently presented in [42] as a novel extension of FHMM-based disaggregation models. First, the time instances where the total energy has more than 60% variation (increase or decrease) are used to break the total energy signal into several energy snippets. Each snippet is assigned to one HMM. In each HMM, the mean of the probability density function corresponding to the hidden states is set to the mean of the observed total energy in the corresponding snippet. Moreover, the transition probability between two HMM states S_i and S_j is proportional to the number of times the snippet corresponding to S_j is observed after the snippet of S_i in the training set. To disaggregate each snippet, each HMM computes the probability of observing the snippet. The snippets corresponding to HMMs with probabilities higher than a threshold $\tau = 0.03$ are considered as the disaggregated signals.

4) Hierarchical FHMM: The HieFHMM is recently presented in [15] for residential ED. In this mode, first, the normalized cross correlation (NCC) of the consumption signals is computed for all devices. Then, the devices are clustered into five groups where each pair of items in each group have an NCC of more than 0.85, while the NCC of items in different groups is less than this threshold. Following the FHMM method in [20], an HMM is trained for each cluster to learn the contribution of each of the five clusters to the total energy.

5) Powerlet-Based Energy Disaggregation: PED is a recent DL model proposed in [29] that led to the state-of-the-art disaggregation performance on the REDD data set. First, the total energy signal is divided into windows of size 15 s. Then, each device is modeled as a set of dynamic systems using autoregression with order 3. For each device, a linear dictionary is learned by ADMM [40] to capture 20 most representative patterns of the consumption signals. To disaggregate the total energy, a linear sparse regression decides which dictionary atoms contribute to the total energy.

E. Numerical Results

Table I shows the ED accuracy computed by (26) for all benchmarks in the REDD data set. As shown in this table, the DL models, PED and DTDL, have generally better performance than other methodologies. FHMM and its variants, AMAPI and HieFHMM, are outperformed by PED and DTDL, since HMM-based models are limited by their first-order Markov property that makes them unable to capture high-order correlation among various devices' consumption patterns. DTDL obtains the highest accuracy with 21.71%, 21.33%, and 18.40% improvement over FHMM, AMAPI, and HieFHMM, respectively. The superiority of DTDL over the benchmarks is due to learning useful nonlinear patterns from electricity signals while incorporating the learned deep features in its DL process. Moreover, the recurrent structure of DTDL makes it a more powerful temporal pattern recognition model for the time-dependent energy data.

TABLE II
PRECISION(%),RECALL(%), AND F-SCORE(%) COMPARISONS

		House					
Method	**Metric**	**1**	**2**	**3**	**4**	**5**	**Average**
SMP	P	37.78	36.52	35.42	36.69	36.73	36.62
	R	35.51	37.64	39.71	42.41	40.75	39.20
	F-score	36.60	37.07	37.44	39.34	38.63	37.82
FHMM	P	77.12	68.81	67.63	71.83	70.09	71.09
	R	53.45	50.02	51.54	54.59	52.88	52.49
	F-score	63.13	57.92	58.49	62.03	60.28	60.37
AMAPI	P	80.56	73.57	75.82	70.27	76.79	75.40
	R	57.83	52.81	55.23	53.29	55.63	54.95
	F-score	67.32	61.48	63.90	60.61	64.51	63.56
HieFHMM	P	80.81	77.02	74.09	67.68	83.01	76.52
	R	58.19	54.85	55.12	52.92	59.91	56.20
	F-score	67.65	64.07	63.21	59.39	69.59	64.78
CDL	P	78.02	71.27	73.58	77.90	89.82	78.11
	R	56.79	53.02	52.11	55.54	56.05	54.70
	F-score	65.73	60.81	61.01	64.84	69.02	64.28
PED	P	86.03	78.89	74.05	77.12	90.02	81.22
	R	62.29	56.70	51.23	55.51	68.22	58.79
	F-score	72.26	65.98	60.56	64.55	77.61	68.19
DTDL	P	**90.95**	**87.31**	**79.60**	**95.27**	**96.01**	**89.83**
	R	**65.12**	**64.03**	**56.18**	**70.12**	**73.45**	**65.78**
	F-score	**75.89**	**73.88**	**65.87**	**80.78**	**83.22**	**75.93**

Table II shows the precision, recall, and F-score of all benchmarks. Let us compare the DL-based models: CDL, PED, and DTDL. On average, DTDL has 18.01% and 11.62% better F-score than CDL and PED, respectively. As explained in Section IV-A, CDL learns a linear dictionary using the consumption signals of all devices. However, PED runs sparse coding on the temporal windows of each device to find the windows that are most representative (i.e., windows that can best represent the whole set of windows). The representative windows of each device are used as the columns of the subdictionary corresponding to that device. In contrast to both CDL and PED, the proposed DTDL algorithm learns a nonlinear dictionary that considers the temporal state transitions of the devices inside each window. DTDL shows better precision and recall than PED and CDL due to modeling the temporal behavior of consumption signals and learning powerful nonlinear features to boost the disaggregation accuracy. The significant superiority of DTDL over CDL and PED shows that the presented deep learning algorithm can better understand the temporal relationships in the REDD data set due to its powerful LSTM-AE features. In addition, it shows that the proposed nonlinear DL optimization in (8) outperforms

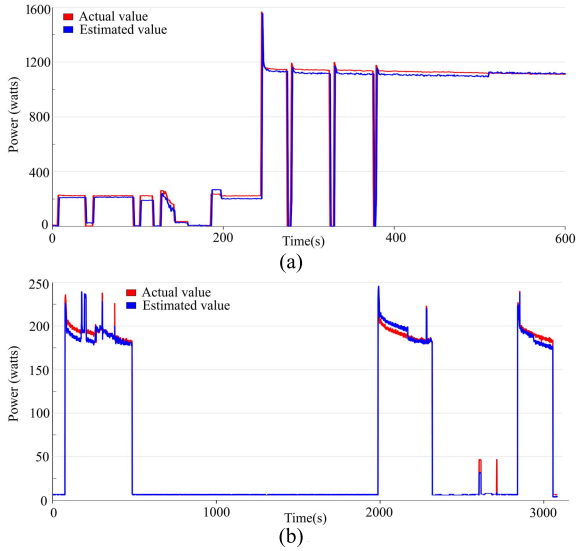


Fig. 6. Estimated energy consumption signals of (a) washer/dryer and (b) refrigerator in House 1 on day 14.

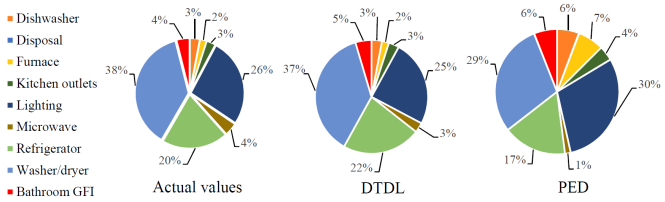


Fig. 7. Pie charts of actual/estimated consumption signals for House 3.

the existing dictionary optimization methods that can merely compute linear dictionaries.

Fig. 6 shows the actual/estimated power consumption obtained by DTDL for two devices in House 1 on day 14 in the testing set. Note that the model accurately understands the transients and various steady states in the appliances. Moreover, Fig. 7 shows the pie charts of the actual/estimated energy consumption of the proposed DTDL and PED for House 3 during the test time. Note that the DTDL's estimated consumption values closely follow the actual values, achieving better accuracy than the state-of-the-art PED model in the 7-day test period. This shows the better reliability of our proposed model for real-world ED purposes in long time horizons.

F. Frequency Resolution Analysis

Section VI-E considers the widely used sampling frequency $f = 1$ Hz to compare the disaggregation performance of DTDL with the recent state-of-the-art benchmarks. In this section, the impact of frequency resolution on the ED performance is studied. As explained in [44], the practical disaggregation methods work with low sampling frequencies in the range $R = [0.2\text{ Hz}, 1\text{Hz}]$. Using frequencies that exceed R would unnecessarily grow the measurement cost while considering frequencies lower than 0.2 Hz would lead to a sensible decline in the measurement accuracy due to the lack of training samples for the disaggregation models.

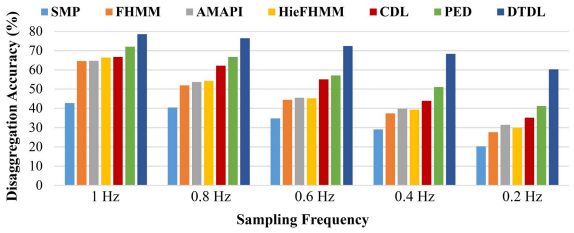


Fig. 8. Average disaggregation accuracy of DTDL and recent benchmarks using various sampling frequencies.

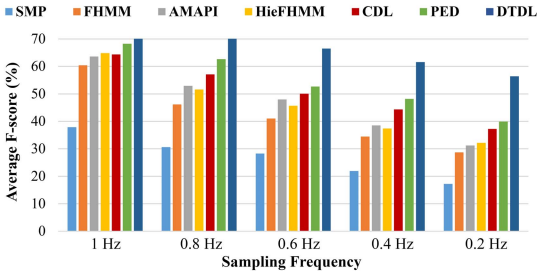


Fig. 9. Average F-score of DTDL and recent benchmarks using various sampling frequencies.

To provide a comprehensive analysis of the effects of sampling frequency resolution on the ED performance, the proposed DTDL is compared with all benchmarks using various values of the empirical frequency $f \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$ in Hz unit. Fig. 8 shows the effect of frequency resolution on the disaggregation accuracy defined in (26) for the proposed DTDL and all benchmarks. As shown in this plot, the accuracy has a positive correlation with the sampling frequency f, that is, the accuracy decreases as the frequency of the electricity signals is decreased. For instance, CDL leads to an accuracy of 66.68% with $f = 1$ Hz, which is decreased to 55.01% for $f = 0.6$ Hz, and reaches to 35.11% when $f = 0.2$ Hz. This observation is due to the fact that using smaller data frequency leads to having less number of training samples that would result in the lower generalization capability of the disaggregation models. On the other hand, increasing the frequency leads to a larger training set that would enhance the generalization of the models. As shown in Fig. 8, DTDL has the lowest decrease rate in comparison with other methods as the frequency is decreased. The accuracy of DTDL drops by only 18.28% from $f = 1$ Hz to $f = 0.2$ Hz, while PED and CDL show 30.80% and 31.57% decrease in the accuracy, respectively. This shows that DTDL can better maintain its generalization capacity when the amount of training data is limited. Fig. 9 shows the impact of sampling frequency in the F-score of DTDL as well as recent disaggregation benchmarks. As shown in this figure, the F-score shows a similar behavior as the disaggregation accuracy when the sampling frequency is changed. The F-score of DTDL declines with a small slope as the frequency is decreased. However, the other methods show a significant decrease in their F-score. Therefore, DTDL provides a more reliable performance in

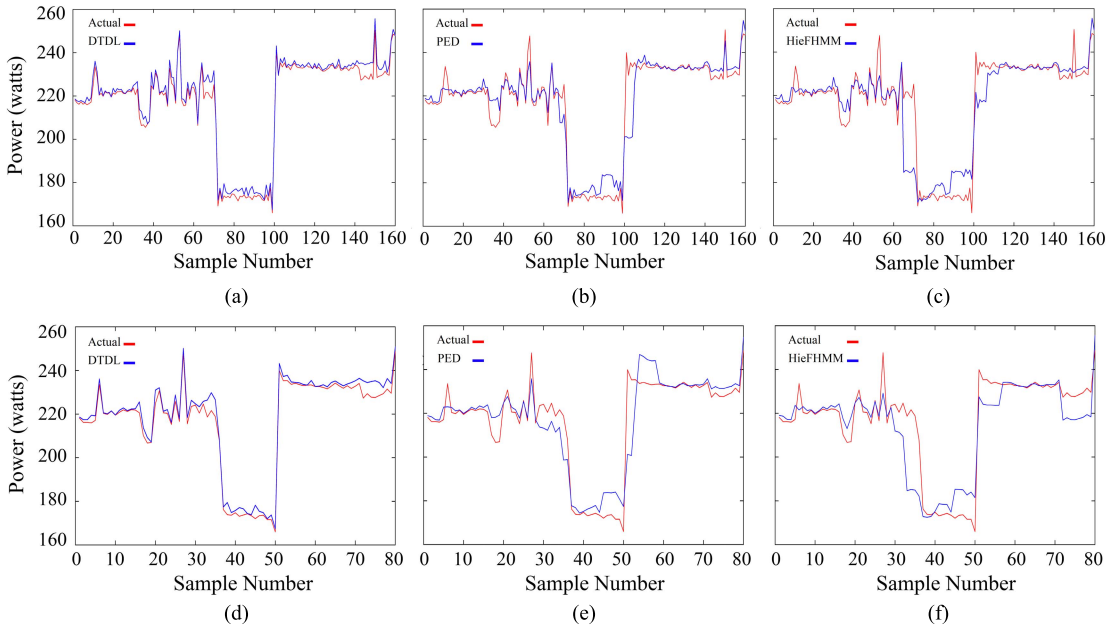


Fig. 10. Estimation of the actual power consumption of furnace in House 3 on day 14 using DTDL, PED, and HieFHMM with $f = 0.8$ Hz and $f = 0.4$ Hz. (a) DTDL with $f = 0.8$ Hz. (b) PED with $f = 0.8$ Hz. (c) HieFHMM with $f = 0.8$ Hz. (d) DTDL with $f = 0.4$ Hz. (e) PED with $f = 0.4$ Hz. (f) HieFHMM with $f = 0.4$ Hz.

real-world applications where the amount of data is limited due to the measurement cost.

Fig. 10 shows the estimation of the actual power consumption of the furnace in House 3 on day 14 using DTDL, PED, and HieFHMM. As shown in Fig. 10(a)–(c), DTDL provides a more accurate estimate of the actual consumption signal than PED and HieFHMM when $f = 0.8$ Hz. DTDL can better estimate the sudden changes in the data including the sudden decrease in sample 72 and the sudden increase in sample 99. The superiority of DTDL over PED and HieFHMM is due to capturing complex nonlinear dictionary atoms from the consumption signals and learning the deep temporal structures and sequential relationships in the data. In Fig. 10(d)–(f), when the frequency is dropped to $f = 0.4$ Hz, DTDL still provides a reliable estimation of the actual power consumption while PED and HieFHMM cannot accurately follow the actual data due to its large variations and nonlinear nature. This observation shows that in contrast to recent benchmarks, DTDL leads to a high disaggregation accuracy even when applied to small data sets with low sampling frequencies and large temporal variations.

VII. Conclusion

In this paper, the problem of ED is addressed as a supervised DL problem. A dictionary matrix is learned to capture the representative consumption patterns of each device, Furthermore, a set of coefficients are optimized to find the most accurate sparse linear combination of these patterns to construct the aggregate electricity signal. To extract informative time-dependent electricity patterns, we propose DTDL that learns deep temporal features from the energy signals of each device using an LSTM-AE. A novel optimization program is devised to learn the LSTM states/parameters while tuning the dictionary atoms and their sparse coefficients using our nonlinear temporal states. Real ED experiments on a publicly available data set show the superiority of the DTDL over HMM-based approaches and DL models. Compared with the state-of-the-art PED, our DTDL obtains 7.63% and 7.10% better disaggregation accuracy and F-score, respectively. This outperformance is mainly due to extracting nonlinear dictionaries as well as learning temporal structure of the underlying electricity signals. Future research seeks to design a new LSTM-AE whose states can be retrieved by an analytical optimizer such as ADMM-based optimization methods to find the global optima temporal parameters.

References

[1] A. Rahimpour, H. Qi, D. Fugate, and T. Kuruganti, "Non-intrusive energy disaggregation using non-negative matrix factorization with sum-to-k constraint," *IEEE Trans. Power Syst.*, vol. 32, no. 6, pp. 4430–4441, Nov. 2017.

[2] R. Arghandeh and Y. Zhou, *Big Data Application in Power Systems*. Amsterdam, The Netherlands: Elsevier, 2018.

[3] S. Gupta, M. S. Reynolds, and S. N. Patel, "ElectriSense: Single-point sensing using EMI for electrical event detection and classification in the home," in *Proc. Conf. Ubiquitous Comput.*, 2010, pp. 139–148.

[4] M. Liu, J. Yong, X. Wang, and J. Lu, "A new event detection technique for residential load monitoring," in *Proc. 18th Int. Conf. Harmon. Qual. of Power (ICHQP)*, 2018, pp. 1–6.

[5] Z. Zhu, S. Zhang, Z. Wei, B. Yin, and X. Huang, "A novel CUSUM-based approach for event detection in smart metering," in *Proc. IOP Conf. Ser., Mater. Sci. Eng.*, vol. 322, Mar. 2018, Art. no. 072014.

[6] I. Rahman, M. Kuzlu, and S. Rahman, "Power disaggregation of combined HVAC loads using supervised machine learning algorithms," *Energy Buildings*, vol. 172, pp. 57–66, Aug. 2018.

[7] Y.-H. Lin and Y.-C. Hu, "Electrical energy management based on a hybrid artificial neural network-particle swarm optimization-integrated two-stage non-intrusive load monitoring process in smart homes," *Processes*, vol. 6, no. 12, p. 236, 2018.

[8] J. Mei and J. M. F. Moura, "Signal processing on graphs: Causal modeling of unstructured data," *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 2077–2092, Apr. 2017.

[9] 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.

[10] 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.

[11] M. A. Mengistu, A. A. Girmay, C. Camarda, A. Acquaviva, and E. Patti, "A cloud-based on-line disaggregation algorithm for home appliance loads," *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 3430–3439, May 2018.

[12] A. Bargi, R. Y. Da Xu, and M. Piccardi, "AdOn HDP-HMM: An adaptive online model for segmentation and classification of sequential data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 3953–3968, Sep. 2018.

[13] Z. Zhihao, L. Jian, Z. Zhen-Yuan, H. Qi, and Q. Hedi, "Non-intrusive load identification methods based on LI-norm and hidden Markov chain model," in *Proc. 2nd IEEE Adv. Inf. Manage., Communicates, Electron. Automat. Control Conf. (IMCEC)*, May 2018, pp. 1875–1879.

[14] 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.

[15] S. El Kababji and P. Srikantha, "Power appliance disaggregation framework via hybrid hidden Markov model," in *Proc. IEEE Can. Conf. Elect. Comput. Eng. (CCECE)*, May 2018, pp. 1–5.

[16] 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.

[17] S. Chen, F. Gao, and T. Liu, "Load identification based on factorial hidden Markov model and online performance analysis," in *Proc. 13th IEEE Conf. Automat. Sci. Eng. (CASE)*, Aug. 2017, pp. 1249–1253.

[18] R. Suzuki, S. Kohmoto, and T. Ogatsu, "Non-intrusive condition monitoring for manufacturing systems," in *Proc. 25th Eur. Signal Process. Conf. (EUSIPCO)*, 2017, pp. 1390–1394.

[19] H. Lange and M. Berges, "Variational bolt: Approximate learning in factorial hidden Markov models with application to energy disaggregation," in *Proc. AAAI*, 2018, pp. 792–799.

[20] R. Bonfigli, E. Principi, M. Fagiani, M. Severini, S. Squartini, and F. Piazza, "Non-intrusive load monitoring by using active and reactive power in additive factorial hidden Markov models," *Appl. Energy*, vol. 208, pp. 1590–1607, Dec. 2017.

[21] J. M. Gillis and W. G. Morsi, "Non-intrusive load monitoring using semi-supervised machine learning and wavelet design," *IEEE Trans. Smart Grid*, vol. 8, no. 6, pp. 2648–2655, Nov. 2017.

[22] A. L. Wang, B. X. Chen, C. G. Wang, and D. Hua, "Non-intrusive load monitoring algorithm based on features of V–I trajectory," *Electr. Power Syst. Res.*, vol. 157, pp. 134–144, Apr. 2018.

[23] Y. Liu, X. Wang, and W. You, "Non-intrusive load monitoring by voltage-current trajectory enabled transfer learning," *IEEE Trans. Smart Grid*, to be published.

[24] K. Khalid, A. Mohamed, R. Mohamed, and H. Shareef, "Nonintrusive load identification using extreme learning machine and TT-transform," in *Proc. Int. Conf. Adv. Elect., Electron. Syst. Eng. (ICAEES)*, 2016, pp. 271–276.

[25] V. M. Salerno and G. Rabbeni, "An extreme learning machine approach to effective energy disaggregation," *Electronics*, vol. 7, no. 10, p. 235, 2018.

[26] X. Zhu, X. Li, S. Zhang, C. Ju, and X. Wu, "Robust joint graph sparse coding for unsupervised spectral feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 6, pp. 1263–1275, Jun. 2017.

[27] T. Shu, B. Zhang, and Y. Y. Tang, "Sparse supervised representation-based classifier for uncontrolled and imbalanced classification," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published.

[28] A. Cherian and S. Sra, "Riemannian dictionary learning and sparse coding for positive definite matrices," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 12, pp. 2859–2871, Dec. 2017.

[29] E. Elhamifar and S. Sastry, "Energy disaggregation via learning powerlets and sparse coding," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 629–635.

[30] C. Mavrokefalidis, D. Ampeliotis, E. Vlachos, K. Berberidis, and E. Varvarigos, "Supervised energy disaggregation using dictionary—Based modelling of appliance states," in *Proc. IEEE PES Innov. Smart Grid Technol. Conf. Eur. (ISGT-Eur.)*, Oct. 2016, pp. 1–6.

[31] A. Majumdar and R. Ward, "Robust dictionary learning: Application to signal disaggregation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 2469–2473.

[32] M. Gupta and A. Majumdar, "Robust supervised sparse coding for non-intrusive load monitoring," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2018, pp. 1–6.

[33] S. Singh and A. Majumdar, "Analysis co-sparse coding for energy disaggregation," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 462–470, Jan. 2019.

[34] F. M. Almutairi, A. Konar, and N. D. Sidiropoulos, "Scalable energy disaggregation via successive submodular approximation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 2676–2680.

[35] M. Khodayar, O. Kaynak, and M. E. Khodayar, "Rough deep neural architecture for short-term wind speed forecasting," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 2770–2779, Dec. 2017.

[36] M. Khodayar, S. Mohammadi, M. E. Khodayar, J. Wang, and G. Liu, "Convolutional graph autoencoder: A generative deep neural network for probabilistic spatio-temporal solar irradiance forecasting," *IEEE Trans. Sustain. Energy*, to be published.

[37] M. Khodayar and J. Wang, "Spatio-temporal graph deep neural network for short-term wind speed forecasting," *IEEE Trans. Sustain. Energy*, vol. 10, no. 2, pp. 670–681, Apr. 2019.

[38] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: Deep belief networks with multitask learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2191–2201, Oct. 2014.

[39] M. Cui, M. Khodayar, C. Chen, X. Wang, Y. Zhang, and M. Khodayar, "Deep learning based time-varying parameter identification for system-wide load modeling," *IEEE Trans. Smart Grid*, to be published.

[40] W. Deng, M.-J. Lai, Z. Peng, and W. Yin, "Parallel multi-block ADMM with $o(1/k)$ convergence," *J. Sci. Comput.*, vol. 71, no. 2, pp. 712–736, 2016.

[41] Z. Zhang, J. He, L. Zhu, and K. Ren, "Non-intrusive load monitoring algorithms for privacy mining in smart grid," in *Advances in Cyber Security: Principles, Techniques, and Applications*. Singapore: Springer, 2018, pp. 23–48.

[42] T. Y. Ji, L. Liu, T. S. Wang, W. B. Lin, M. S. Li, and Q. H. Wu, "Non-intrusive load monitoring using additive factorial approximate maximum *a posteriori* based on iterative fuzzy C-means," *IEEE Trans. Smart Grid*, to be published.

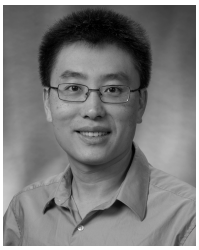
[43] A. Terenin, S. Dong, and D. Draper, "GPU-accelerated Gibbs sampling: A case study of the Horseshoe Probit model," *Statist. Comput.*, vol. 29, no. 2, pp. 301–310, 2018.

[44] C. Shin, S. Rho, H. Lee, and W. Rhee, "Data requirements for applying machine learning to energy disaggregation," *Energies*, vol. 12, no. 9, p. 1696, 2019.

Mahdi Khodayar (S'17) received the B.Sc. degree in computer engineering and the M.Sc. degree in artificial intelligence from the Khajeh Nasir Toosi University of Technology, Tehran, Iran, in 2013 and 2015, respectively. He is currently pursuing the Ph.D. degree in electrical engineering with Southern Methodist University, Dallas, TX, USA.

He was a Research Assistant with the College of Computer and Information Science, Northeastern University, Boston, MA, USA, in 2017. His current research interests include machine learning, statistical pattern recognition, deep learning, sparse modeling, and spatiotemporal pattern recognition.

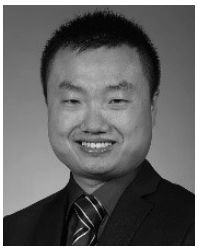
Mr. Khodayar has served as a Reviewer for reputable journals including the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the IEEE TRANSACTIONS ON FUZZY SYSTEMS, the IEEE TRANSACTIONS ON SUSTAINABLE ENERGY, and the IEEE TRANSACTIONS ON POWER SYSTEMS.



Jianhui Wang (M'07-SM'12) received the Ph.D. degree in electrical engineering from the Illinois Institute of Technology, Chicago, IL, USA, in 2007.

He had an 11-year stint with the Argonne National Laboratory, Lemont, IL, USA, as the Section Lead of advanced grid modeling. He has held visiting positions in Europe, Australia, and Hong Kong including a VELUX Visiting Professorship with the Technical University of Denmark (DTU). He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Southern Methodist University, Dallas, TX, USA.

Dr. Wang is the Editor-in-Chief of the IEEE TRANSACTIONS ON SMART GRID and an IEEE PES Distinguished Lecturer. He is the Secretary of the IEEE Power & Energy Society (PES) Power System Operations, Planning, & Economics Committee. He is also a Clarivate Analytics Highly Cited Researcher for 2018.



Zhaoyu Wang (S'13-M'15) received the B.S. and M.S. degrees in electrical engineering from Shanghai Jiaotong University, Shanghai, China, in 2009 and 2012, respectively, and the M.S. and Ph.D. degrees in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2012 and 2015, respectively.

He was a Research Aid with the Argonne National Laboratory, Lemont, IL, USA, in 2013. He was an Electrical Engineer Intern with Corning Inc., Corning, NY, USA, in 2014. His current research interests include power distribution systems, microgrids, renewable integration, power system resilience, and data-driven system modeling. He is currently the Harpole-Pentair Assistant Professor with Iowa State University, Ames, IA, USA. He is also the Principal Investigator for a multitude of projects focused on these topics and funded by the National Science Foundation, the Department of Energy, National Laboratories, PSERC, and Iowa Energy Center.

Dr. Wang is an Editor of the IEEE TRANSACTIONS ON POWER SYSTEMS, the IEEE TRANSACTIONS ON SMART GRID, and IEEE PES LETTERS, and an Associate Editor of *IET Smart Grid*. He is the Secretary of IEEE Power and Energy Society Award Subcommittee.