IEEE Open Journal of the Computer Society

Received 10 July 2025; revised 25 August 2025; accepted 6 September 2025. Date of publication 11 September 2025; date of current version 24 September 2025. The review of this article was arranged by Associate Editor Lorenzo Bottaccioli.

Digital Object Identifier 10.1109/OJCS.2025.3609195

Multivariate Constrained Elastic Matching With Application in Real-Time Energy Disaggregation

PASCAL A. SCHIRMER, DIMITRIOS KOLOSOV, AND IOSIF MPORAS

[1]University of Hertfordshire, AL10 9AB Hatfield, U.K.

CORRESPONDING AUTHOR: PASCAL A. SCHIRMER (e-mail: p.schirmer@herts.ac.uk).

ABSTRACT Non-Intrusive Load Monitoring (NILM) aims to estimate the power consumption of electrical appliances from the aggregated power consumption. While recent machine learning approaches have demonstrated very high disaggregation accuracies, ensuring real-time capability is crucial in NILM's hardware implementations. We propose a constrained elastic matching approach for NILM to reduce execution time significantly. Our approach was tested on two datasets (REDD and AMPds2). The reported performance is on average 93.2% in terms of estimation accuracy for deferrable loads using the AMPds2 dataset. The proposed approach reduces execution time by a factor of ten compared to unconstrained elastic matching techniques, achieving per-frame inference times of 3.5–12.1 ms depending on the hardware platform and model size. Memory usage for the largest model is approximately 7.5 MB, and reducing the model to 10% of reference signatures lowers active power consumption from 12.1 W to 5.2 W, representing a 57% energy saving with minimal accuracy loss. Furthermore, the proposed approach has been evaluated on five different microprocessors, demonstrating consistent runtime reduction and enabling real-time implementation of elastic matching based NILM with large reference databases.

INDEX TERMS Energy disaggregation, non-intrusive load monitoring (NILM), smart meter, smart grid, consumer households, pattern matching, elastic matching, dynamic time warping (DTW).

NOMENCLATURE

A	Alignment matrix for elastic matching.
D	Number of features.
E	Total energy consumption (kWh).
F	Dimensionality of the reduced order feature space.
I	RMS current (A).
k	Model size (%).
L	Frame/Window length.
N_r	Number of reference signatures.
N_t	Number of test signatures.
P	Active power (W).
Q	Reactive power (VAr).
S	Apparent power (VA).
T	Number of samples.
T_s	Sampling time (sec).
W	Reference signature database.
X_{agg}	Aggregated total smart meter signal.
c_g	Global alignment cost.
c_l	Local alignment cost.
e	Measurement noise.
p_{agg}	Aggregated active power (W).
p_m	Active power of an appliance (W).
t_{EM}	Computational time for EM algorithm (sec).
x^τ	One frame of x with index τ.
$\hat{x}$	Predicted value of x.
$\tilde{x}$	Reduced feature space of x.
$\delta(\cdot)$	Distance metric, e.g. Euclidean distance.
ϵ	Small error margin.
$f(\cdot)$	Aggregation function.
$f^{-1}(\cdot)$	Estimation function.
μ_{train}	Mean value of the training data.
σ_{train}	Standard deviation of the training data.
$\mathcal{T}$	Total number of frames.
$v(\cdot)$	Set of statistical function, e.g. mean, max, etc.
m, n, i	Index variables.
ACC	Accuracy.

E_{ACC}	Estimation accuracy.
EM	Elastic Matching.
F1	F1 accuracy score.
MAE	Mean Absolute Error.
MBW	Memory BandWidth.
RMSE	Root Mean Square Value.
SAE	Signal Aggregated Error.

I. INTRODUCTION

Non-Intrusive Load Monitoring aims to extract the power consumption at the device level only from the aggregated power consumption signal at the inlet of a household or building [1]. Therefore, NILM offers a cost-effective and scalable solution for energy consumption monitoring in buildings, providing a valuable data source for smart grid implementations while preserving consumer privacy [2], [3], [4], [5]. The NILM or energy disaggregation task can be formulated as a single-channel source separation problem [6]. Three main families of approaches have been proposed to solve this problem, namely, those based on Machine Learning (ML), Pattern Matching (PM), and Source Separation (SS) [7].

Specifically, ML-based approaches, due to their ability to model non-linear dependencies and under-determined problems, have been used extensively to address the NILM problem. Early approaches have focused mostly on Hidden Markov Models (HMMs) [8] and their variants [9]. In recent years, Long-Short-Term Memory (LSTM) [10], [11] and Convolutional Neural Networks (CNNs) [12] have been investigated due to their ability to model temporal information and multivariate signatures [13], respectively. The most recent approaches focus on Generative Adversarial Networks (GANs) [14], [15], [16], Denoising Auto Encoder (DAE) [17], and bidirectional Transformers [18] to incorporate self-attention mechanisms and to further improve the performance of energy disaggregation. Since NILM is intrinsically a source separation problem, SS techniques like Non-Negative Matrix Factorization (NMF) [6] and Discriminative Sparse Coding (DSC) [19] have been used for energy disaggregation, while the latest proposed methods have utilized multiple features [20] or considered the temporal content and operation of many appliances at the same time [21], [22]. The advantage of these techniques is that they are unsupervised by the nature of the corresponding algorithms; however, they rely on a priori information, making them semi-unsupervised. However, PM-based techniques have been used to exploit appliance signatures that can be observed in the aggregated signal under the superposition of other appliances' signatures. Therefore, Dynamic Time Warping (DTW) [23] and other Elastic Matching (EM) algorithms, e.g., Multi Variance Matching (MVM), Global Alignment Kernel (GAK), or soft Dynamic Time Warping (sDTW), have been utilized to identify appliance signatures from the aggregated signal [24]. Similarly to the SS-based methods, pattern matching does not rely on a trained model, but on a set of reference pattern signatures stored in a database.

The development of large datasets available for NILM [25], [26] and the improvements in Graphical Processing Units (GPUs) have enabled the efficient training of machine learning-based approaches from large amounts of collected energy data as well as on high-frequency data ($\gg$ 1Hz) [12], thus machine learning based approaches dominate the NILM task [12], [27], [28]. However, given the increasing need for running NILM on edge devices (fully or partially) [29], the energy disaggregation algorithms must work with limited hardware resources. Therefore, the transfer learning-based methods [30], [31] and the utilization of pre-trained models [32] or training-less approaches [33] have been exploited most recently, and other NILM approaches have also been investigated to work with very low sampling frequencies [34] or to reduce latencies within the NILM architecture [35]. Moreover, scalable and light-weight solutions based on low sampling frequencies and CNNs have been proposed to achieve real-time NILM [36], [37].

Given that pattern matching-based approaches have not benefited from parallel computing on GPUs, few recent NILM approaches based on pattern matching have been proposed, despite their promising results [24], [38]. Especially, as they do not perform well with large amounts of data due to long monitoring durations or high sampling frequencies. We propose a multivariate constrained Elastic Matching (cEM) algorithm that overcomes the computational burdens of pattern-matching-based NILM approaches. The contribution is threefold: First, an elastic matching-based NILM architecture is proposed that achieves on average performance at the state-of-the-art compared to the best-reported machine learning-based models, while showing a significant reduction of execution time and not relying on model training. Specifically, the reduction in computational load of the proposed method is theoretically derived based on the formulation of overall algorithm complexity. It enables for the first time the usage of pattern matching based approaches in real-time. Second, the approach is evaluated on five different microprocessors, demonstrating runtime advantages on hardware applications, enabling real-time capability, and further optimizing NILM models according to hardware restrictions. The practical evaluation validates the theoretically estimated reduction in computational complexity of the proposed method across all evaluated hardware setups and demonstrates that real-time usage of elastic matching based NILM is possible using the proposed method. Third, an exhaustive comparison with machine learning and source separation methods is provided, focusing on accuracy, scalability, transferability, runtime, and memory requirements. It is shown that the proposed constrained elastic matching approach is advantageous in terms of runtime, while achieving comparable results in the other performance criteria using only 10% of the data compared to the original elastic matching approach.[1]

[1] The approach is integrated here: https://github.com/pascme05/BaseNILM

The remainder of the article is structured as follows: In Section II, the proposed cEM algorithm and the corresponding NILM architecture are introduced. In Section III, the experimental setup is provided. In Section IV, the experimental results are presented. Discussion is provided in Section V, and the article is concluded in Section VI.

II. CONSTRAINED ELASTIC MATCHING FOR NILM

Let $X_{agg} \in \mathbb{R}^{T\times D}$ be the aggregated signal acquired by a smart meter, where T is the number of samples and D is the number of features, which can be active power (P), reactive power (Q), apparent power (S), and current (I). Furthermore, let $X_a^\tau \in \mathbb{R}^{L\times D}$ and $X_b^\tau \in \mathbb{R}^{L\times D}$ be two frames a and b of X_{agg}, where L is the frame length and τ is the frame index. Let $\Delta(X_a^\tau, X_b^\tau) = [\delta(x_i^a, x_j^b)]_{i,j} \in \mathbb{R}^{L\times L}$ be an arbitrary cost matrix evaluating the quality of the alignment of an elastic matching algorithm, where $\delta(\cdot)$ is a distance metric, such as the Euclidean, the Manhattan, or the Kullback Leibler distance, and i, j denote the sample indices in the two frames X_a^τ and X_b^τ. Based on the definition of the inner product $\langle A, \Delta(X_a^\tau, X_b^\tau)\rangle$, where A is the alignment matrix of the elastic matching algorithm with $A \in \mathbb{R}^{L\times L}$ giving the scores of A and the inner product is defined as $\langle A, \Delta\rangle = \sum_{1\le i,j\le} a_{i,j}\delta_{i,j}$, the cost of all possible alignments for the EM can be written as:

$$EM\left(X_a^\tau, X_b^\tau\right) = \min_{A\in A_{L,L}} \langle A, \Delta(X_a^\tau, X_b^\tau)\rangle \tag{1}$$

where EM can be any elastic matching algorithm, e.g., DTW, sDTW, GAK, or MVM.

A. CONSTRAINED ELASTIC MATCHING (CEM)

Assuming a database of N_r reference signatures $W : W_n,\ 1 \le n \le N_r$ with $W_n \in \mathbb{R}^{L\times D}$ and N_t test signatures, there would be $N = N_t \cdot N_r$ reference-test signature pair distance estimations necessary to find the reference-test pair with the best matching, i.e., the minimum distance. As the computational complexity of EM is in the order of $\mathcal{O}(n^2)$, the approach is unsuitable for real-time applications if N is large. For instance, considering a signature database containing data from one year of recordings with a sampling rate of $T_s = 60$ sec and one-sample overlap between successive frames, would result in $N_r^{\text{year}} = 365 \cdot 24 \cdot 60 \approx 0.5$ million reference signatures. To enable real-time energy disaggregation, the computation of (1) must be performed significantly faster than $\frac{T_s}{N_r^{\text{year}}} \approx 114\,\mu$s, to accommodate additional computational overhead and ensure real-time processing. Previous evaluations have shown that disaggregation can be performed in the order of milliseconds up-to 1 ms [8], thus not fast enough for real-time implementation of classical EM algorithms. For low-cost hardware, the disaggregation time per frame should ideally be even shorter.

However, since typically only a few signatures of the reference dataset are close to the test signature, the number of EM searches can be reduced. In detail, let $v(\cdot)$ be a feature mapping function that transfers the time domain input signature X^τ to a set of low-frequency statistical features $\tilde{X}^\tau \in \mathbb{R}^F$ with dimensionality $F \ll (L \times D)$. Similarly, an element of the dataset of reference signatures W_n can be transformed using $v(\cdot)$, i.e., $\tilde{W}_n \in \mathbb{R}^F$, where $v(\cdot)$ is a set of statistical functions, e.g., mean, variance, max/min, et cetera. Subsequently, the number of EM searches can be reduced by restricting the search to the reference signatures that are close to the test signature in the feature space, i.e.:

$$\left\|\tilde{W} - \tilde{X}^\tau\right\|_p \le \varepsilon \tag{2}$$

where $\|\cdot\|$ denotes the L_p-norm with $p \ge 1$ and ε is an error margin defining the maximum required distance in the feature space and in turn the size of the reduced set of reference signatures to be processed.

The computational cost of the EM algorithm in (1), applied to two frames $X_a^\tau, X_b^\tau \in \mathbb{R}^{L\times D}$, is $\mathcal{O}(L^2D)$ in time and $\mathcal{O}(L^2)$ in memory due to the need to compute the pairwise distance matrix Δ and perform dynamic programming for all L^2 alignment paths. For a dataset of N_r reference signatures, this results in a total time complexity of $\mathcal{O}(N_rL^2D)$ per test signature, which is prohibitive for real-time applications. To address this, (2) introduces a constrained search strategy that projects each signature into a lower-dimensional feature space $\mathbb{R}^F$ (with $F \ll L \times D$), allowing a fast feature domain reduction of the search space of the reference signatures with time complexity $\mathcal{O}(N_rF)$ and negligible memory requirement. This reduces the number of costly EM signature alignments from N_r to $kN_r \ll N_r$, where $k \ll 1$ is the effective model size (in percent) that depends on the error margin ε. This results in an improved overall complexity of $\mathcal{O}(N_rF + kL^2D)$ per test signature, with significant gains in time usage when F and k are small. The theoretical improvement of computational complexity is described in (3):

$$\frac{t_{cEM}}{t_{EM}} \propto \frac{N_rF + kN_rL^2D}{N_rL^2D} = \frac{F}{L^2D} + k \approx k \tag{3}$$

where t_{cEM} is the computational time of the proposed constrained EM algorithm and t_{EM} is the computational time of the original EM algorithm. The pipeline for reducing the candidate reference signatures in the database is illustrated in Fig. 1.

As shown in Fig. 1, the process consists of four steps. First, recording the reference signature database W in the time domain results in N_r reference signatures of size $(L \times D)$. Second, low-frequency features which create the feature vector dataset $\widetilde{W}$. Third, finding the closest samples in the feature space using an error margin ε for each unknown signature $\widetilde{X}$. Fourth, reducing the number of reference signatures based on the constraint in (2), resulting in the dataset of reduced reference signatures $W' \in \mathbb{R}^{N_r'\times(L\times D)}$ with N_r' signatures and $N_r' \ll N_r$. The constraint elastic matching algorithm is described in Algorithm 1 below:

B. PROPOSED NILM ARCHITECTURE

NILM aims to determine the device-level power consumption based on measurements from a single aggregating sensor within a specified time window (frame). Specifically, for a

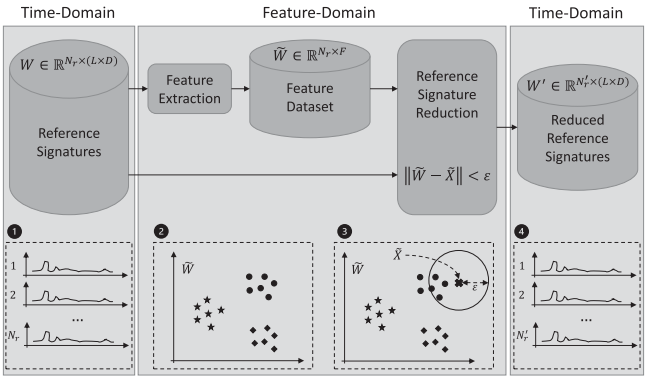

FIGURE 1. Feature domain-based signature database reduction for the cEM algorithm, including reference signature recording, feature extraction, feature matching, and reference signature reduction.

Algorithm 1: Constrained Elastic Matching.

Input: NILM reference signature database W; Test sample of aggregated frame X^{τ}_{agg}
Output: Index of closest match n'
Initialization: Local align cost $c_l \rightarrow \infty$; Global align cost $c_g \rightarrow \infty$

```
for signature n in W do
    W̃_n → generate signature feature vector v(W_i)
    X̃^τ → generate aggregated feature vector v(X^τ_agg)
    if (‖W̃_n − X̃^τ‖_p ≤ ε) then
        c_l → evaluate local cost EM(W^agg_n, X^τ_agg)
        if c_l < c_g then
            c_g → c_l updating global cost
        end if
    end if
    n' → assign index with to smallest global cost c_g
end for
return n'
```

set of $M-1$ known devices each of them consuming power $p_m \in \mathbb{R}^T$, with $1 \leq m \leq M-1$ and T being the total number of samples, the aggregated power $p_{agg} \in \mathbb{R}^T$, measured by the single sensor will be:

$$p_{agg} = f(p_1, \ldots, p_{M-1}, e) = \sum_{m=1}^{M-1} p_m + e = \sum_{m=1}^{M} p_m \quad (4)$$

where $e = p_M \in \mathbb{R}^T$ is noise generated by one or more unknown devices and $f(\cdot)$ is the aggregation function. In NILM the goal is to find estimations $\hat{p}_m, \hat{e}$ of the power consumption of each device m using an estimation method $f^{-1}(\cdot)$ with minimal estimation error and $\hat{p}_M = \hat{e}$, i.e.:

$$\hat{P} = \{\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_{M-1}, \hat{e}\} = f^{-1}(p_{agg}) \quad (5)$$

As (5) is practically impossible to solve analytically, most energy disaggregation methodologies segment the aggregated signal into frames and estimate the power consumption based on features.

$$\hat{P}^{\tau} = \{\hat{p}^{\tau}_1, \hat{p}^{\tau}_2, \ldots, \hat{p}^{\tau}_{M-1}, \hat{e}^{\tau}\} = f^{-1}(X^{\tau}_{agg}) \quad (6)$$

where $\hat{P}^{\tau}$ is the predicted power consumption of the $M-1$ appliances and the noise for the τ^{th} frame. To solve the disaggregation problem using an elastic matching algorithm during training, a signature database $W : W_n$, $1 \leq n \leq N_r$ is created using both the aggregated and appliances signals, i.e, $W_n = [W^{agg}, W^1, W^2, \ldots, W^M]$. During testing, the difference between an unknown signature X^{τ}_{agg} and the reference signature database W is then calculated using the cEM algorithm as formulated in Algorithm 1. The closest match between an unknown signature X^{τ}_{agg} and the reference signature database W can then be written as in (7), while the estimates of the appliance power consumptions can be determined as in (8):

$$n'(\tau) = \text{argmin}_{n \in N_r} \left\{ cEM\left(X^{\tau}_{agg}, W^{agg}_n\right) \right\} \quad (7)$$

$$\hat{P}^{\tau} = \left\{ \frac{1}{L}\sum_{L} W^1_{n'(\tau)}, \frac{1}{L}\sum_{L} W^2_{n'(\tau)}, \ldots, \frac{1}{L}\sum_{L} W^M_{n'(\tau)} \right\} \quad (8)$$

where $cEM(\cdot)$ is the constraint elastic matching algorithm and $n'(\tau)$ is the index of the closest match of the signature in the database. Since $W^m_{n'(\tau)}$ is a matrix of size $L \times D$, the average of $\frac{1}{L}\sum_L W^m_{n'(\tau)}$ is computed across the axis including the active power of the m-th device to predict $\hat{p}^{\tau}_m$.

TABLE 1. Short Description of the REDD and AMPds2 Datasets. *P*, *Q*, *S*, and *I* are the Active Power, Reactive Power, Apparent Power, and RMS Current, Respectively. T_s is the Sampling Period

Name	Duration	#-Houses	#-Devices	Features	T_s
REDD	23 – 48 d	6	9 – 24	P	3 s
AMPds2	2 y	1	20	P, Q, S, I	60 s

III. EXPERIMENTAL SETUP

The NILM architecture based on the proposed cEM algorithm in Section II was evaluated using the datasets, parametrization, and features presented below.

A. DATASETS

Since NILM requires a previously recorded dataset for the appliance models, the proposed architecture was evaluated using two different datasets, namely the REDD [39] and the AMPds2 [26]. The REDD dataset is the most used in the energy disaggregation task, thus allowing direct comparison with other methods proposed in the literature. In contrast, the AMPds2 dataset provides multiple features and does not require pre-processing for two years of recordings. A description of the datasets is shown in Table 1.

In the NILM literature, either all appliances are used or a subset of them [8], usually referred to as deferrable loads, i.e., loads that can be used at a different time. Thus, our results are presented for both setups. The deferrable loads

TABLE 2. Optimization of the Frame Length in Terms of Estimation Accuracy E_{ACC} (12) Using All Appliances of the REDD Dataset

Data	Framelength L				
	5	10	15	20	25
REDD-1	75.31%	**75.86%**	75.85%	74.84%	74.47%
REDD-2	76.07%	**76.40%**	76.05%	73.43%	73.32%
REDD-3	**60.24%**	59.73%	59.74%	60.02%	59.92%
REDD-4	59.86%	59.75%	60.67%	**60.93%**	60.47%
REDD-6	72.93%	73.61%	**73.89%**	73.68%	73.66%
All	68.88%	69.07%	**69.24%**	68.58%	68.37%

TABLE 3. Optimization of the Distance Metric in Terms of Estimation Accuracy E_{ACC} (12) Using All Devices of the REDD Dataset

Data	Distance Metric				
	EUC	MIN	MAN	COS	HAM
REDD-1	**75.85%**	75.85%	75.85%	68.99%	72.57%
REDD-2	**76.05%**	76.05%	76.05%	68.96%	67.70%
REDD-3	**59.74%**	59.74%	59.74%	55.85%	57.77%
REDD-4	**60.67%**	60.67%	60.67%	58.36%	60.03%
REDD-6	**73.89%**	73.89%	73.89%	71.08%	73.48%
All	**69.24%**	69.24%	69.24%	64.65%	66.31%

for the REDD dataset are the kettle, the microwave, the dishwasher, the fridge, and the washing machine [9], while for the AMPds2 dataset are the clothes dryer (CDE), the dishwasher (DWE), the HVAC system (FRE), the heat pump (HPE), and the kitchen wall oven (WOE) [8].

B. PRE-PROCESSING AND PARAMETRIZATION

Samples normalization has shown performance improvements, especially in transfer learning [40], thus during pre-processing, mean-variance scaling was applied to the input feature vectors as described in (9):

$$x' = \frac{x - \mu_{train}}{\sigma_{train}} \tag{9}$$

where μ_{train} is the mean value of the input values x during training, σ_{train} is their standard deviation, and x' is the mean-variance scaled version of x. Furthermore, the free parameters of the algorithm, namely the frame length, the distance metric, and the restriction on the warping path were optimized using a bootstrap training dataset (first five days of each house of REDD) and five-fold cross-validation. During pre-processing optimizations, DTW was used as the EM algorithm. The frame length and the distance metric optimization results are tabulated in Tables 2 and 3.

As can be seen in Table 2 the optimal frame length was found to be 15 samples averaged across five houses of the REDD dataset, while Euclidean (EUC), Minkowski (MIN), and Manhattan (MAN) distance metrics have achieved the same performance, outperforming Cosine (COS) and Hamming (HAM) distance metrics, thus Euclidean distance has been selected in all following experiments, as it is the most computationally efficient one. Furthermore, different restrictions on the warping path have been tested, namely Sakoe and Itakura [41], [42]. The restriction on the warping path reports significantly worse performance, which is in agreement with [24], thus no restriction has been applied to the warping path.

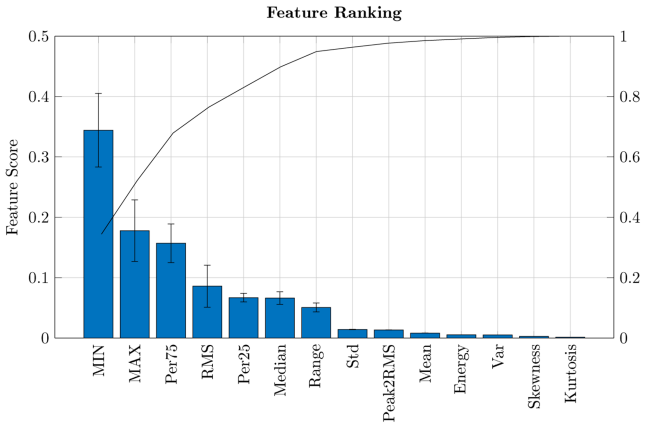

FIGURE 2. Relief-F based feature ranking for 14 different statistical low-frequency features, including the cumulative feature score (black line) and the standard deviation.

C. FEATURE SELECTION

To reduce the model size described in Section II-A, a set of low-frequency statistical features is calculated from the frames of the aggregated signal X^{τ}_{agg}. Feature ranking was conducted using ReliefF [43] and the low-frequency features proposed in [44]. The results of the feature ranking are shown in Fig. 2.

As can be seen in Fig. 2, approximately 95% of the ReliefF-based information can be found in the first seven features (Minimum, Maximum, 75% Percentile, Root-Mean Square value, 25% Percentile, Median, and Range), thus in the experimental results below only these seven features have been used, and discussion on the influence of the feature selection on the performance is provided in Subsection V-B.

D. HARDWARE

The real-time performance of the proposed cEM algorithm, as well as its ability to reduce the runtime and memory requirements of the proposed approach, was evaluated using various hardware configurations. The hardware configurations, including their most relevant parameters, are tabulated in Table 4.

The hardware platforms used are a mixture of various ARM Cortex-A application processors with different external memory configurations (density/data bus speed/bus width). Although some platforms offer acceleration capabilities (GPU, FPGA, etc.), only their CPU resources were evaluated. The

TABLE 4. Overview of Considered Hardware Configurations. In the First Column, the Name in Brackets is the Name Used in the Text and Figures

Platform	CPU	RAM	Freq.	Bus	MBW	$/Unit
Raspberry PI 4 (RPI-4)	Cortex-A72 (1.50GHz)	4 GB LPDDR4	3200 MHz	32 bit	12.8 GB/s	55
Jetson Nano (Jetson)	Cortex-A57 (1.43GHz)	4 GB LPDDR4	3200 MHz	64 bit	25.6 GB/s	99
KV260	Cortex-A53 (1.33GHz)	4 GB DDR4	1200 MHz	64 bit	9.6 GB/s	249
i.MX 8M Plus EVK (iMX)	Cortex-A53 (1.80GHz)	6 GB LPDDR4	2133 MHz	32 bit	8.5 GB/s	449
Ultra96v1 (Ultra96)	Cortex-A53 (1.20GHz)	2 GB LPDDR4	533 MHz	32 bit	2.1 GB/s	249

maximum theoretical Memory Bandwidth (MBW) was also calculated to assess the external memory performance of each platform, as defined in (10).

$$MBW = \frac{Clock\ Frequency \times Bus}{8} \tag{10}$$

IV. EXPERIMENTAL RESULTS

The architecture presented in Section II was evaluated according to the experimental setup described in Section III. NILM performance was evaluated in terms of estimation accuracy (E_{ACC}), as proposed in [39], i.e.

$$E_{ACC} = 1 - \frac{\sum_{\tau=1}^{\mathcal{T}} \sum_{m=1}^{M} |\hat{p}_m^\tau - p_m^\tau|}{2 \sum_{\tau=1}^{\mathcal{T}} \sum_{m=1}^{M} |p_m^\tau|} \tag{11}$$

where p_m^τ and $\hat{p}_{m_\tau}$ are the real and estimated power consumption (closest match from (8)), respectively, of the m-th device at the τ-th frame, $\mathcal{T}$ is the number of disaggregated frames and M is the number of disaggregated devices. Furthermore, to compare with other approaches previously published in the literature, additional accuracy metrics, namely the Mean Absolute Error (MAE) and the normalized Signal Aggregated Error (SAE), were used:

$$MAE = \frac{1}{\mathcal{T}} \sum_{\tau=1}^{\mathcal{T}} |\hat{p}_m^\tau - p_m^\tau| \tag{12}$$

$$SAE = \frac{|\hat{E}^m - E^m|}{E^m} \tag{13}$$

where E_m denotes the total energy consumption of the m-th appliance and $\hat{E}_m$ its predicted value.

A. UNIVARIATE CEM BASED ENERGY DISAGGREGATION

For the experimental setup based on univariate data, i.e., when only one feature is available ($D = 1$), the REDD dataset has been utilized as it includes only active power measurements. In detail, the data were pre-processed as described in Section III-B and downsampled to one sample per minute for a fair comparison with the literature [19]. The frame length was 15 samples, with 14 overlapping samples between successive frames. For simplicity, from here onwards, the number of reference signatures used is denoted as the percentage of the total size of each dataset and will be referred to as model size. The total number of reference signatures, i.e., model size equal to 100%, for each dataset used in this article, is tabulated in Table 5.

TABLE 5. Equivalent Number of Reference Signatures (In Thousands) for Different Datasets for a Model Size of 100%

REDD-1	REDD-2	REDD-3	REDD-4	REDD-6	AMPds2
24k	18k	22k	25k	17k	950k

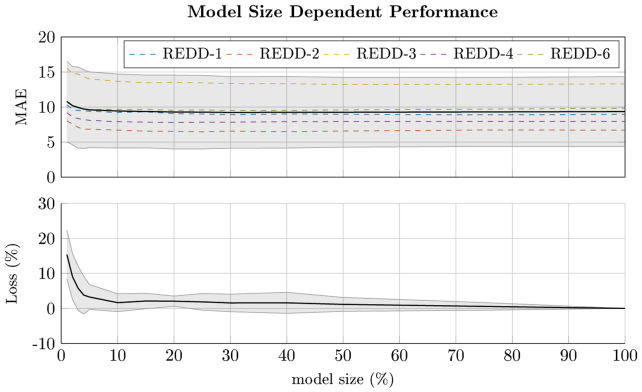

FIGURE 3. Univariate cEM NILM results for REDD (all loads) in terms of MAE for different numbers of reference signatures (model size (%)). Solid lines indicate average performance and dashed lines indicate the performance of the houses (REDD-1: blue, REDD-2: red, REDD-3: yellow, REDD-4: purple, and REDD-6: green). The gray areas indicate the 2-σ confidence interval.

The results in terms of MAE and relative performance drop (loss) compared to using all the reference signatures, i.e., unconstrained EM, are illustrated in Fig. 3. In detail, the loss is the difference of the MAE between constrained and unconstrained EM, normalized by the MAE of the unconstrained EM, and expressed as a percentage.

As can be seen in Fig. 3, the proposed cEM algorithm's performance in terms of MAE varies between 15.6 – 13.3 (REDD-3) and 8.0 – 6.7 (REDD-2) depending on the number of reference signatures. In detail, a significant increase of the MAE can be observed for all houses when reducing the number of reference signatures below 2 k (10%), with the error increasing by 1.6% on average. The MAE increases up to 15% when using only 200 (1%) of the reference signatures with a two-sigma confidence interval of ±8.4%.

B. MULTIVARIATE CEM BASED ENERGY DISAGGREGATION

To evaluate the performance when multiple features ($D = 4$) are available, the AMPds2 dataset was used. In detail, the data were pre-processed as described in Section III-B, and the NILM feature setups from [28] have been used to ensure a fair comparison. The frame length was chosen to be 30 samples (30 minutes). The results for both deferrable and all loads when using single-fold validation, with 90% training and 10% testing, are tabulated in Tables 6 and 7. DTW was used as the EM algorithm.

TABLE 6. Results for AMPds2 (All Loads) in Terms of E_{ACC} Using Different Model Sizes

Features	model size (%)				
	0.1%	0.5%	1%	5%	10%
P (Out: P)	67.20%	70.90%	71.68%	72.98%	**72.99%**
I (Out: I)	72.53%	75.42%	76.14%	76.80%	**76.83%**
Q (Out: Q)	77.58%	80.78%	81.41%	**81.69%**	81.68%
S (Out: S)	74.16%	77.22%	77.44%	78.32%	**78.61%**
PQ (Out: P)	74.21%	76.43%	77.00%	77.38%	**77.42%**
All (Out: P)	75.08%	77.43%	77.93%	78.11%	**78.46%**

TABLE 7. Results for AMPds2 (Deferrable Loads) in Terms of E_{ACC} Using Different Model Sizes

Features	model size (%)				
	0.1%	0.5%	1%	5%	10%
P (Out: P)	78.41%	83.45%	84.19%	85.47%	**85.49%**
I (Out: I)	81.75%	85.83%	86.62%	**86.88%**	86.87%
Q (Out: Q)	83.14%	86.82%	**87.63%**	87.14%	86.84%
S (Out: S)	82.23%	86.47%	86.30%	87.15%	**87.53%**
PQ (Out: P)	86.13%	88.50%	89.00%	**89.10%**	88.92%
All (Out: P)	86.27%	88.61%	**89.04%**	88.92%	89.27%

As shown in Tables 6 and 7, the reactive power (Q) and the apparent power (S) are achieving the best performances followed by the RMS current (I) and the active power (P) when using only one feature. Furthermore, using multivariate features, e.g., the reactive and active power (PQ), the performance is further improved when disaggregating active power. When using all features (ALL), the best performance is observed regardless of the number of reference signatures. Disaggregating reactive power has the highest performance values, as it is an easier disaggregation problem with resistive appliances having a reactive power consumption close to zero; this is in line with the results reported in [28], [45]. Moreover, a similar trend as in the univariate results shown in Section IV-A can be observed for the multivariate results, i.e. the performance drops significantly when using less than 9.5k (1%) reference signatures (it must be noted that AMPDs2 has two years of data, thus far fewer data is required), while only minor performance differences can be observed when using more than 47.5k reference signatures (5%). It should be noticed that when disaggregating all loads, the benefit of having more data is more significant compared to only disaggregating the deferrable loads.

C. ELASTIC MATCHING

As discussed in [24] alongside DTW, other elastic matching algorithms have shown significant performance improvements for NILM due to their ability to skip outliers, for MVM [46], or through utilizing smoothing of the warping path, for sDTW or GAK [47]. Therefore, three additional elastic matching approaches are evaluated as cEM functions using the AMPds

TABLE 8. Performance Comparison for Different Accuracy Metrics and Elastic Matching Algorithms. The Free Parameter Used for the Kernel of GAK is $\sigma = 2000$, for the Soft Alignment of sDTW is $\gamma = 0.5$, and for the Step-Width of MVM is $step = 10$

Mdl	ACC	F1	E_{ACC}	MAE	RMSE	SAE
DTW	**97.88%**	**97.83%**	92.91%	0.11	0.94	0.003
sDTW	97.84%	97.79%	92.72%	0.11	0.96	0.001
GAK	97.71%	97.67%	93.31%	0.10	0.83	**0.001**
MVM	97.83%	97.77%	**93.55%**	**0.10**	**0.79**	0.004

TABLE 9. Performance Comparison for Different Accuracy Metrics and Appliances for MVM

App	ACC	F1	E_{ACC}	MAE	RMSE	SAE
DWE	97.06%	97.03%	46.03%	0.13	0.88	0.02
FRE	99.85%	99.82%	94.35%	0.15	0.26	0.00
HPE	92.84%	92.60%	95.82%	0.13	1.01	0.01
WOE	99.56%	99.54%	89.62%	0.03	0.83	0.06
CDE	99.84%	99.84%	95.48%	0.04	0.97	0.03

TABLE 10. Literature Comparison With Previously Proposed Approaches

Method	REF	Loads	Metric	REF	MVM
WaveNILM	[28]	Def	E_{ACC}	**94.7%**	93.2%
SSFHMM	[8]	Def (noisy)	E_{ACC}	**94.1%**	93.6%
SSFHMM	[8]	Def (denoised)	E_{ACC}	98.1%	**98.2%**
EnerGAN	[15]	HPE, WOE, COE	RMSE	221.0	**166.7**
EnerGAN++	[14]	HPE, WOE, COE	RMSE	176.4	**166.7**
BabiLSTM	[27]	HPE, WOE, COE	RMSE	**158.6**	166.7

dataset (first year of AMPds2). In detail, all input features (RMS current as the output), a fixed number of reference signatures of 5%, a constant frame length of ten samples, and 10-fold cross-validation have been used. To be comparable with as many approaches as possible, results are presented using six different accuracy metrics, including next to (11)– (13), also Root-Mean-Square-Error (RMSE), accuracy (ACC), and F1-scores (F1). The results are tabulated in Table 8.

As tabulated in Table 8, sDTW achieves roughly identical performance, while GAK and MVM outperform the conventional DTW algorithm, showing performance improvements of 0.4 – 0.6% in terms of E_{ACC}. MVM reports the best performance for three out of five accuracy metrics. Furthermore, the results on the appliance level are tabulated in Table 9.

As tabulated in Table 9, the heat pump has the best disaggregation accuracy according to the E_{ACC} metric, and the dishwasher has the worst performance among the selected appliances. Additionally, the best-performing NILM approaches reported in the literature are tabulated in Table 10 and are compared to the proposed cEM algorithm using 5% of all reference signatures. black WaveNILM [28] and SSFHMM [8] have been explicitly selected for three reasons. First, their source code is publicly available, allowing a direct one-to-one

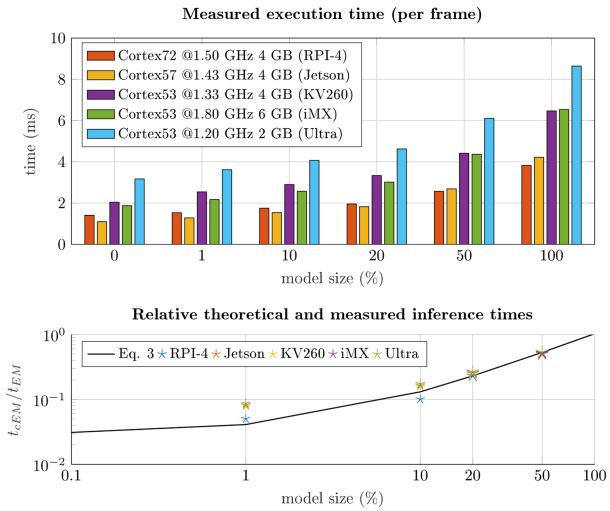

FIGURE 4. Results of the per-sample hardware inference times using the REDD-2 (all loads) database. The upper sub-figure shows the absolute execution time per frame, and the lower sub-figure shows the relative inference time for different model sizes. In the lower subfigure, the black solid line indicates the theoretical value from (3), while the stars indicate the measured values.

comparison. Second, both methods use the AMPds dataset, which does not require any pre-processing since it does not contain any missing or implausible values or time gaps in the measurements, making it the dataset with the lowest ambiguity in comparisons. Third, their reported performance is obtained using 10-fold cross-validation, ensuring a fair and standardized comparison.

As can be seen in Table 10, the proposed cEM algorithm reports results at the state-of-the-art, only 0.5 – 1.5% lower than the best performing WaveNILM approach [28], while it even slightly outperforms previously proposed machine learning approaches like HMMs and GANs.

D. RUNTIME, ENERGY CONSUMPTION, AND MEMORY

In Fig. 4, the hardware performances are shown and compared in terms of absolute (upper subfigure) and relative (lower subfigure) execution time per disaggregated frame for different model sizes. A model size of 100% refers to using an unconstrained EM approach, while a model size of k% corresponds to cEM using only the k% of the reference signatures of the database W.

As shown in Fig. 4, the absolute execution times vary per hardware platform. Model size, CPU type, CPU clock frequency, and external memory specifications all contribute to the execution of this application, meaning the fastest CPU clock or faster memory frequency does not necessarily correlate to the fastest execution speed. In detail, the results are similar from 1% to 20% of the model size (dominated by the computational overhead), but for 50% RPI-4 outperforms Jetson even though it has half of Jetson's memory bandwidth. For 100% model size, KV260 outperforms iMX by a small margin. However, when calculating the relative measured inference time of the cEM as $(t_{k=x} - t_{k=0})/t_{k=100}$, where k is the model size, it can be seen (lower subfigure of 4) that the results are in line with the theoretical inference time from (3).

TABLE 11. Measured Standby, Maximum, and Active Power Consumption (Watts) During Inference for Different Model Sizes Across Hardware Platforms

Board	Standby	Max	Model Size (%)				
			1	10	20	50	100
RPI-4	3.3	9.0	3.5	3.7	4.1	5.4	8.7
Jetson	3.1	10.0	3.4	3.5	3.9	5.1	8.2
KV260	6.8	20.0	7.3	7.7	8.5	11.2	18.0
iMX	3.6	15.0	4.1	4.1	4.5	5.9	9.5
Ultra	6.1	24.0	6.8	7.0	7.7	10.0	16.1
Avg	4.6	15.6	5.0	5.2	5.7	7.5	12.1

The size of the reduced signature database W' does not depend on the utilized hardware. For the data from the utilized REDD-2 dataset, which contains approximately 18,000 reference signatures (model size of 100%) of sequence length $L = 15$ and feature dimensionality $D = 1$, the memory requirement is $18,000 \cdot 15 \cdot 10 \cdot 2\,bytes \approx 5.1$ MB for $M = 9$ appliances and the aggregated signal samples using FP16 representation. Similarly, $\tilde{W}$ does not depend on the utilized hardware or the model size, and the memory requirement for $F = 7$ features is $18.000 \cdot 7 \cdot 10 \cdot 2\,bytes \approx 2.4$ MB for an FP16 data representation, resulting into a total memory requirement of 7.5 MB.

In addition to runtime and memory usage, the energy consumption of the hardware platforms was also measured under both standby and active inference conditions. Table 11 summarizes the power draw (in Watts) across the five evaluated boards for model sizes varying from 1% to 100%.

As can be seen in Table 11, the active power consumption scales significantly with model size: inference using 100% of the model leads to an average draw of 12.1 W, compared to just 5.2 W at 10% model size. This represents a reduction of nearly 57%, with only a marginal loss in inference quality for EM methods as shown in Tables 6 and 7. Assuming 24 hours of daily operation, reducing the model size from 100% to 10% would save approximately 165.6 Wh per day, or around 60.4 kWh annually.

V. DISCUSSION

Further, to the experimental results presented in Section IV, the comparison with ML and SS-based methods and the influence of features and the number of reference signatures on execution time and memory requirements are investigated for the proposed constrained elastic matching.

A. COMPARISON WITH MACHINE LEARNING AND SOURCE SEPARATION

As discussed in [7], three fundamentally different NILM approaches have been utilized: machine learning, pattern matching, and source separation. Each has advantages and

TABLE 12. Comparative Evaluation of All Loads for the REDD-2 Dataset for ML, Constrained PM, and SS Approaches

Metric	ML		PM		SS	
	CNN	LSTM	cDTW	cMVM	NMF	DSC
ACC	97.98%	97.92%	98.54%	**98.58%**	76.82%	89.54%
F1	97.89%	97.78%	98.54%	**98.56%**	75.32%	91.91%
E_{ACC}	87.13%	87.30%	90.14%	**91.09%**	40.29%	43.71%
MAE	4.41	4.34	3.30	**3.00**	27.60	18.78
SAE	0.033	0.036	**0.009**	0.016	0.165	0.155

TABLE 13. Comparative Evaluation of Deferrable Loads for the REDD-2 Dataset for ML, Constrained PM, and SS Approaches

Metric	ML		PM		SS	
	CNN	LSTM	cDTW	cMVM	NMF	DSC
ACC	96.33%	96.78%	97.02%	**97.11%**	66.04%	85.26%
F1	95.58%	96.55%	96.95%	**97.04%**	68.07%	87.44%
E_{ACC}	89.12%	90.38%	92.36%	**92.51%**	47.38%	43.21%
MAE	5.73	5.03	5.11	**4.99**	48.70	29.60
SAE	0.043	**0.002**	0.031	0.035	0.218	0.306

TABLE 14. Comparative Evaluation of All the Transferability for the REDD-2 Dataset (Trained on REDD-1) for ML, Constrained PM, and SS Approaches

Metric	ML		PM		SS	
	CNN	LSTM	cDTW	cMVM	NMF	DSC
ACC	81.82%	**81.95%**	75.70%	75.68%	72.83%	69.88%
F1	77.28%	**77.43%**	71.56%	71.52%	74.07%	75.68%
E_{ACC}	38.74%	**39.38%**	33.46%	33.33%	37.31%	12.92%
MAE	32.51	**32.17**	46.18	46.28	33.22	46.14
SAE	0.628	0.681	0.440	0.440	0.245	**0.226**

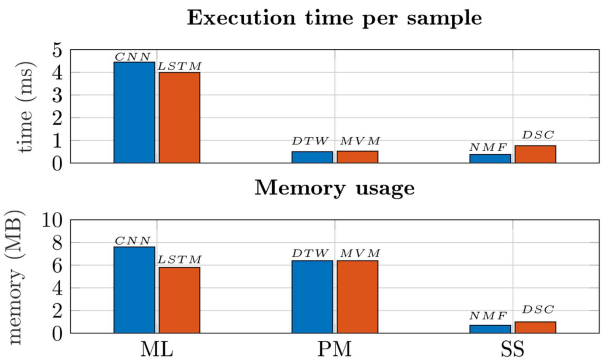

FIGURE 5. Comparison of ML, PM, and SS methods considering execution time per sample for training/testing, and memory requirements.

limitations and can be compared based on five criteria: accuracy, runtime, memory requirements, scalability, and transferability [7]. The following provides a comparative evaluation of ML, PM, and SS techniques, considering two specific models, i.e., CNN and LSTM for ML, constrained cDTW and cMVM for PM, and NMF and DSC for SS. The five criteria are evaluated as described below, and have been chosen such that lower values are always better for each of the five categories:

1) *Accuracy:* Is evaluated in terms of MAE using all appliances
2) *Runtime:* Is evaluated in terms of execution time (sec) for training and testing
3) *Memory:* Is evaluated in terms of the model size (MB) of the trained model
4) *Scalability:* Is evaluated in terms of the accuracy difference between all loads and deferrable loads
5) *Transferability:* Is evaluated in terms of MAE when training and testing are done on different houses

For the CNN and LSTM, standard model structures have been implemented. The CNN consists of five 1D-CNN layers with the number of filters being equal to [30, 30, 40, 50, 50] and kernel sizes being equal to [10, 8, 6, 5, 5], followed by three fully connected DNN layers with 256 nodes each. The LSTM consists of two LSTM layers with 128 nodes each, followed by three fully connected DNN layers with 256 nodes each. The NMF [6] and DSC [9] approaches have been re-implemented from the literature. For all evaluations, REDD-2 has been used to calculate results using 10-fold cross-validation (using 10% of the training data for validation). For evaluating the transferability, the model was trained using REDD-1, while testing was conducted using REDD-2. The data was not further down-sampled, and a restriction of 1% on the number of reference signatures has been applied for the constrained PM approaches. The results of the accuracy evaluation for the deferrable, all loads, and the transferability setup are shown in Tables 12, 13, and 14.

Similarly, runtime and memory requirements are evaluated on a desktop PC running an AMD Ryzen 3700X, as shown in Fig. 5 for ML, PM, and SS approaches, respectively.

To compare the results, a rating of the five criteria accuracy, runtime, memory, scalability, and transferability is shown in Fig. 6. In detail, for each of the three methods (ML, PM, and SS), the average of the two implemented approaches, e.g., CNN and LSTM for ML, has been used for evaluation. Then the relative score is calculated by dividing the average score of each approach by the sum of the scores of all approaches and taking the inverse, resulting in a relative rating between zero (worst) and one (best).

As shown in Fig. 6, PM and ML outperform SS in terms of accuracy. This is due to the excellent capability of neural networks to model the non-linear relationships between device signatures and the aggregated signal, and PM pattern-matching techniques capture the signal envelope for different operation states. Furthermore, ML also reports the best result for transferability, followed by SS and PM. This is due to the ability of a neural network to generalize appliance signatures, while SS and PM are based on dictionaries. Conversely, ML is showing the worst results for both execution time and memory requirements due to the training period of the neural network and the number of weights that need to be stored in the model. Vice versa, SS is reporting the best performance in terms of

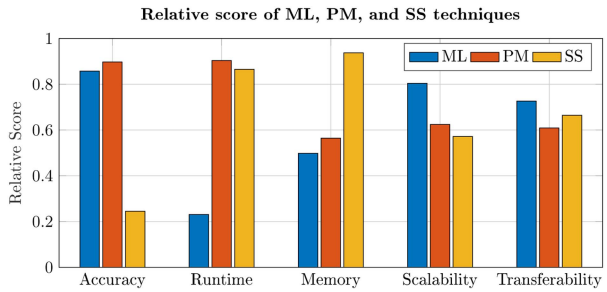

FIGURE 6. Relative comparison of ML, PM, and SS methods for five different criteria, values close to zero denote worse performance and values close to one superior performance.

execution time and memory due to formulating the disaggregation problem as a linear algebra problem. Lastly, machine learning shows the best performance in terms of scalability, operating as an automated feature extraction engine. Given that PM does not perform best in terms of transferability, memory requirement, and scalability, further investigation on these metrics was performed:

Transferability: In terms of transferability, the difference between the relative scores of PM, ML, and SS is small compared to accuracy, runtime, memory or scalability metrics, with the transferability results being in a range of ±5% (ML = 71%, PM = 61%, SS = 65%). Consequently, a detailed statistical analysis with additional data would be required to quantify the influence of the three methods on transferability; this is outside the scope of this article. However, possible approaches to improve transferability include investigating more normalization techniques to homogenize the signature database, e.g. a quantile transformation, which has shown promising results in [48], or enhancing the reference signatures reduction process by selecting more distinctive features or exploiting physical correlations between them, such as the relationship $S = \sqrt{P^2 + Q^2}$ for apparent power when using multi-input features as in AMPds2 [49]. It may also be worthwhile to use the disaggregation obtained from the proposed technique as a pre-prediction stage for a second machine-learning model, following the concept presented in [50].

Memory: While the memory requirements of the ML and PM approaches are comparable in magnitude, their underlying causes differ fundamentally. In the ML approach, the memory footprint is primarily determined by the large number of trained model parameters, which collectively define the model size. These parameters must be loaded into memory simultaneously to compute a prediction from a given input feature vector. In contrast, the memory requirement of the proposed PM approach is governed by the size of the signature database W. When a new input feature vector is received, the database W is first reduced to a subset W' based on similarity criteria, and the subsequent disaggregation is carried out on this reduced set. Based on the previous evaluations, a size of W' of around 1-10% of W is sufficient without suffering a significant loss in disaggregation accuracy.

Scalability: Although ML approaches (CNN, LSTM) show the best relative scalability as indicated by the smallest performance gap between all loads and deferrable loads, the absolute disaggregation accuracy achieved by PM methods remains the highest across both scenarios. This implies that while ML maintains more consistent performance across different appliance types, PM still yields superior overall results, albeit with slightly greater relative degradation when focusing on deferrable loads. The higher variability in PM performance is largely due to the greater challenge of matching variable or overlapping usage patterns for certain load types, particularly when using a constrained signature database. Nonetheless, the drop in PM performance remains moderate and acceptable in light of its strong absolute accuracy. Future work could explore adaptive or hierarchical matching strategies that dynamically adjust signature matching criteria based on appliance characteristics or load context. Another promising direction is to incorporate confidence-based selection schemes or ensemble models that leverage both PM and ML outputs [51]. However, these enhancements are beyond the scope of this article.

B. FEATURES AND MODEL SIZE

As described in Subsection II-A, the cEM algorithms have two free parameters influencing their performance. First, the restriction on the number of reference signatures (model size) is extensively evaluated in terms of performance in the previous chapters. Second, the number of features used to perform the model order reduction. These two parameters are independent but mutually influence performance, execution time, and memory requirements. Therefore, a grid search was used to investigate their impact on the three requirements. In detail, the REDD dataset was used, and the number of reference signatures varied between 1–10%, while the feature dimensionality was successively increased according to their ReliefF ranking scores as calculated in Section III-C (Fig. 2). The results are illustrated in Fig. 7.

As shown in Fig. 7(a), performance improves when adding additional features and increasing the number of reference signatures. Specifically, as shown in the feature ranking in Fig. 2, the most significant performance improvement is observed when using the three to five top-ranked features; using more than seven features shows only a minor performance improvement. Similarly, the largest performance improvement is observed when using up to 3% of the available reference signatures. Furthermore, as shown in Fig. 7(b), runtime increases roughly linearly both when adding additional features and when increasing the number of reference signatures, which is in agreement with the theoretical discussions in Section II-A. Moreover, memory increases linearly with a linear increase in reference signatures and with a linear increase in the number of features. The optimal number of features and the maximum possible reference signatures can be optimized for a given hardware configuration, i.e., based on the available memory and processing power. An example of a hardware restriction

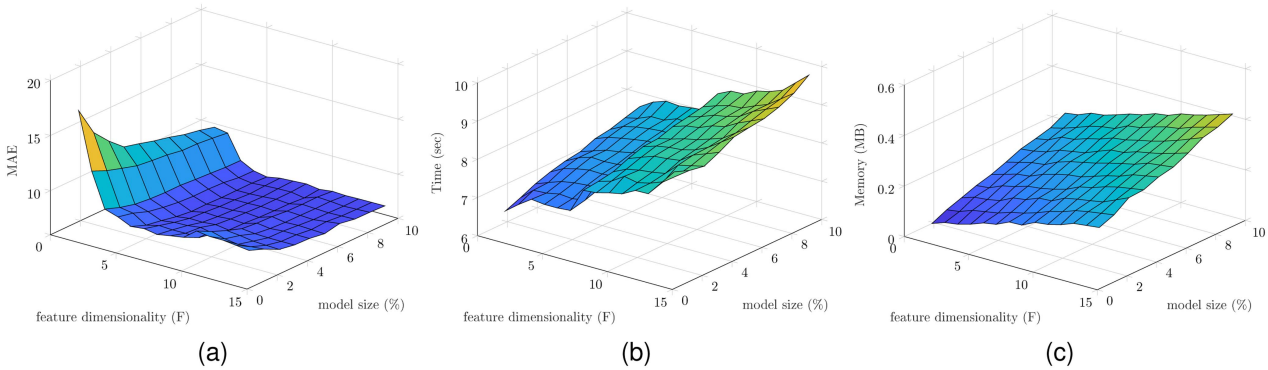

FIGURE 7. Influences of model size and feature dimensionality of the cDTW algorithm on (a) performance, (b) execution time, and (c) memory usage.

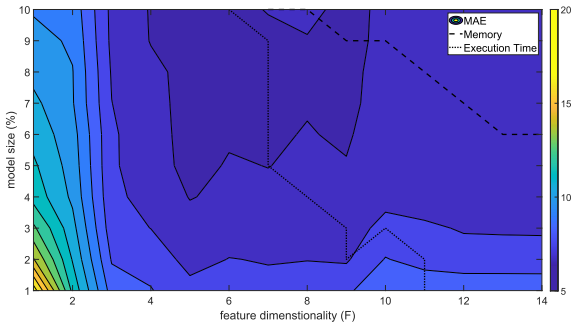

FIGURE 8. Restriction on available memory (0.3 MB) and execution time (8 sec) for a given hardware application.

requiring a model size smaller than 0.3 MB and an execution time of less than 8 seconds is illustrated in Fig. 8.

In Fig. 8, the two restrictions on memory and execution time are illustrated using dashed (memory) and dotted lines (execution time). In detail, the areas above these lines are not possible under the given hardware restrictions.

VI. CONCLUSION

A constrained elastic matching algorithm with application to energy disaggregation has been proposed, enabling real-time energy disaggregation using elastic matching-based approaches. The proposed algorithm can drastically reduce execution time while achieving performances close to state-of-the-art compared to the best-performing approaches in the literature. In detail, it was shown that reducing the number of reference signatures by a factor of ten does not significantly influence performance. For larger datasets, reductions as low as 1% of the original data size might be feasible. Furthermore, the algorithms were tested using five different microprocessor architectures, demonstrating the real-time capability for hardware implementations. Moreover, it was shown that the algorithm enables optimal disaggregation for given hardware applications by adapting runtime and memory requirements independently. The authors hope the results will increase research interest in pattern-matching-based energy disaggregation approaches due to their advantages for hardware applications. The following two aspects should be considered in future research: First, the robustness of electrical coupling or noise on the electrical lines. In a first step, this could be evaluated by artificially amplifying the noise signal e or adding additional Gaussian White Noise to the aggregated data. For further evaluations, the coupling between different devices and the impact of on/off transitions should be investigated using high-frequency measurement data. Second, an in-depth evaluation of transfer learning scenarios should be conducted, considering the absolute number of reference signatures for each operating state of the appliances.

REFERENCES

[1] G. W. Hart, "Nonintrusive appliance load monitoring," *Proc. IEEE*, vol. 80, no. 12, pp. 1870–1891, Dec. 1992.
[2] C.-C. Sun, D. J. S. Cardenas, A. Hahn, and C.-C. Liu, "Intrusion detection for cybersecurity of smart meters," *IEEE Trans. Smart Grid*, vol. 12, no. 1, pp. 612–622, Jan. 2021.
[3] Y. Tao, J. Qiu, S. Lai, Y. Wang, and X. Sun, "Reserve evaluation and energy management of micro-grids in joint electricity markets based on non-intrusive load monitoring," *IEEE Trans. Ind. Appl.*, vol. 59, no. 1, pp. 207–219, Jan./Feb. 2023.
[4] H. Wang, J. Zhang, C. Lu, and C. Wu, "Privacy preserving in non-intrusive load monitoring: A differential privacy perspective," *IEEE Trans. Smart Grid*, vol. 12, no. 3, pp. 2529–2543, May 2021.
[5] P. A. Schirmer and I. Mporas, "On the non-intrusive extraction of residents' privacy-and security-sensitive information from energy smart meters," *Neural Comput. Appl.*, vol. 35, pp. 119–132, 2023.
[6] 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.
[7] P. A. Schirmer and I. Mporas, "Non-intrusive load monitoring: A review," *IEEE Trans. Smart Grid*, vol. 14, no. 1, pp. 769–784, Jan. 2023.
[8] S. Makonin, F. Popowich, I. V. Bajic, B. Gill, and L. Bartram, "Exploiting HMM sparsity to perform online real-time nonintrusive load monitoring," *IEEE Trans. Smart Grid*, vol. 7, no. 6, pp. 2575–2585, Nov. 2016.
[9] J. Z. Kolter and T. Jaakkola, "Approximate inference in additive factorial HMMs with application to energy disaggregation," in *Proc. 15th Int. Conf. Artif. Intell. Statist.*, La Palma, Canary Islands, 2012, pp. 1472–1482. [Online]. Available: http://proceedings.mlr.press/v22/zico12.html
[10] T.-T.-H. Le, S. Heo, and H. Kim, "Toward load identification based on the hilbert transform and sequence to sequence long short-term memory," *IEEE Trans. Smart Grid*, vol. 12, no. 4, pp. 3252–3264, Jul. 2021.
[11] M. Kaselimi, N. Doulamis, A. Voulodimos, E. Protopapadakis, and A. Doulamis, "Context aware energy disaggregation using adaptive bidirectional LSTM models," *IEEE Trans. Smart Grid*, vol. 11, no. 4, pp. 3054–3067, Jul. 2020.

[12] P. A. Schirmer and I. Mporas, "Double fourier integral analysis based convolutional neural network regression for high-frequency energy disaggregation," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 6, no. 3, pp. 439–449, Jun. 2022.
[13] J. Chen, X. Wang, X. Zhang, and W. Zhang, "Temporal and spectral feature learning with two-stream convolutional neural networks for appliance recognition in NILM," *IEEE Trans. Smart Grid*, vol. 13, no. 1, pp. 762–772, Jan. 2022.
[14] M. Kaselimi, N. Doulamis, A. Voulodimos, A. Doulamis, and E. Protopapadakis, "EnerGAN++: A generative adversarial gated recurrent network for robust energy disaggregation," *IEEE Open J. Signal Process.*, vol. 2, pp. 1–16, 2021.
[15] M. Kaselimi, A. Voulodimos, E. Protopapadakis, N. Doulamis, and A. Doulamis, "EnerGAN: A generative adversarial network for energy disaggregation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 1578–1582.
[16] A. Harell, R. Jones, S. Makonin, and I. V. Bajić, "TraceGAN: Synthesizing appliance power signatures using generative adversarial networks," *IEEE Trans. Smart Grid*, vol. 12, no. 5, pp. 4553–4563, Sep. 2021.
[17] D. Garcia-Perez, D. Perez-Lopez, I. Diaz-Blanco, A. Gonzalez-Muniz, M. Dominguez-Gonzalez, and A. A. C. Vega, "Fully-convolutional denoising auto-encoders for NILM in large non-residential buildings," *IEEE Trans. Smart Grid*, vol. 12, no. 3, pp. 2722–2731, May 2021.
[18] Z. Yue, C. R. Witzig, D. Jorde, and H.-A. Jacobsen, "BERT4NILM: A bidirectional transformer model for non-intrusive load monitoring," in *Proc. 5th Int. Workshop Non-Intrusive Load Monit.*, 2020, pp. 89–93.
[19] J. Z. Kolter, S. Batra, and A. Y. Ng, "Energy disaggregation via discriminative sparse coding," in *Proc. 24th Int. Conf. Neural Inf. Process. Syst.*, 2010, pp. 1153–1161. [Online]. Available: http://papers.nips.cc/paper/4054-energy-disaggregation-via-discriminative-sparse-coding
[20] P. A. Schirmer and I. Mporas, "Multivariate non-negative matrix factorization with application to energy disaggregation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 3285–3289.
[21] G. Tanoni, E. Principi, and S. Squartini, "Multilabel appliance classification with weakly labeled data for non-intrusive load monitoring," *IEEE Trans. Smart Grid*, vol. 14, no. 1, pp. 440–452, Jan. 2023.
[22] R. Jiao, C. Li, G. Xun, T. Zhang, B. B. Gupta, and G. Yan, "A context-aware multi-event identification method for nonintrusive load monitoring," *IEEE Trans. Consum. Electron.*, vol. 69, no. 2, pp. 194–204, May 2023.
[23] B. Liu, W. Luan, and Y. Yu, "Dynamic time warping based non-intrusive load transient identification," *Appl. Energy*, vol. 195, pp. 634–645, 2017.
[24] P. A. Schirmer, I. Mporas, and M. Paraskevas, "Energy disaggregation using elastic matching algorithms," *Entropy*, vol. 22, no. 1, 2020, Art. no. 71.
[25] D. Murray, L. Stankovic, and V. Stankovic, "An electrical load measurements dataset of United Kingdom households from a two-year longitudinal study," *Sci. Data*, vol. 4, no. 1, pp. 1–12, 2017.
[26] 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, 2016.
[27] M. Kaselimi, N. Doulamis, A. Doulamis, A. Voulodimos, and E. Protopapadakis, "Bayesian-optimized bidirectional LSTM regression model for non-intrusive load monitoring," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2019, pp. 2747–2751.
[28] A. Harell, S. Makonin, and I. V. Bajic, "WaveNILM: A causal neural network for power disaggregation from the complex power signal," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2019, pp. 8335–8339.
[29] Y. Zhang et al., "FedNILM: Applying federated learning to NILM applications at the edge," *IEEE Trans. Green Commun. Netw.*, vol. 7, no. 2, pp. 857–868, Jun. 2023.
[30] P. A. Schirmer and I. Mporas, "Device and time invariant features for transferable non-intrusive load monitoring," *IEEE Open Access J. Power Energy*, vol. 9, pp. 121–130, 2022.
[31] J. Lin, J. Ma, J. Zhu, and H. Liang, "Deep domain adaptation for non-intrusive load monitoring based on a knowledge transfer learning network," *IEEE Trans. Smart Grid*, vol. 13, no. 1, pp. 280–292, Jan. 2022.
[32] L. Wang, S. Mao, B. M. Wilamowski, and R. M. Nelms, "Pre-trained models for non-intrusive appliance load monitoring," *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 1, pp. 56–68, Mar. 2022.
[33] A. Majumdar, "Trainingless energy disaggregation without plug-level sensing," *IEEE Trans. Instrum. Meas.*, vol. 71, 2022, Art. no. 2504808.
[34] S. Singh and A. Majumdar, "Multi-label deep blind compressed sensing for low-frequency non-intrusive load monitoring," *IEEE Trans. Smart Grid*, vol. 13, no. 1, pp. 4–7, Jan. 2022.
[35] Y. Liu, J. Qiu, and J. Ma, "SAMNet: Toward latency-free non-intrusive load monitoring via multi-task deep learning," *IEEE Trans. Smart Grid*, vol. 13, no. 3, pp. 2412–2424, May 2022.
[36] C. L. Athanasiadis, T. A. Papadopoulos, and D. I. Doukas, "Real-time non-intrusive load monitoring: A light-weight and scalable approach," *Energy Buildings*, vol. 253, 2021, Art. no. 111523.
[37] C. Athanasiadis, D. Doukas, T. Papadopoulos, and A. Chrysopoulos, "A scalable real-time non-intrusive load monitoring system for the estimation of household appliance power consumption," *Energies*, vol. 14, no. 3, 2021, Art. no. 767.
[38] 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.
[39] J. Z. Kolter and M. J. Johnson, "Redd: A public data set for energy disaggregation research," in *Proc. Workshop Data Mining Appl. Sustainability*, San Diego, CA, USA, 2011, pp. 59–62.
[40] M. DrIncecco, S. Squartini, and M. Zhong, "Transfer learning for non-intrusive load monitoring," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1419–1429, Mar. 2020.
[41] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *IEEE Trans. Acoust. Speech Signal Process.*, vol. TASSP-23, no. 1, pp. 67–72, Feb. 1975.
[42] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. TASSP-26, no. 1, pp. 43–49, Feb. 1978.
[43] N. Spolaôr, E. A. Cherman, M. C. Monard, and H. D. Lee, "Relieff for multi-label feature selection," in *Proc. Braz. Conf. Intell. Syst.*, 2013, pp. 6–11.
[44] P. A. Schirmer and I. Mporas, "Statistical and electrical features evaluation for electrical appliances energy disaggregation," *Sustainability*, vol. 11, no. 11, 2019, Art. no. 3222.
[45] P. A. Schirmer and I. Mporas, "Energy disaggregation using fractional calculus," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 3257–3261.
[46] L. J. Latecki et al., "Elastic partial matching of time series," in *Proc. Knowl. Discov. Databases: PKDD 2005*, A. M. Jorge, L. Torgo, P. Brazdil, R. Camacho, and J. Gama, Eds., Berlin, Germany, 2005, pp. 577–584.
[47] M. Cuturi and M. Blondel, "Soft-DTW: A differentiable loss function for time-series," 2017, in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2017.
[48] I. U. Khan, A. Ali, C. J. Taylor, and X. Ma, "Data-driven insights: Boosting algorithms to uncover electricity theft patterns in AMI," *IEEE Trans. Instrum. Meas.*, vol. 74, 2025, Art. no. 2524212.
[49] G. Huang, Z. Zhou, F. Wu, and W. Hua, "Physics-informed time-aware neural networks for industrial nonintrusive load monitoring," *IEEE Trans. Ind. Informat.*, vol. 19, no. 6, pp. 7312–7322, Jun. 2023.
[50] Q. Li, X. Zhang, T. Ma, D. Liu, H. Wang, and W. Hu, "A multi-step ahead photovoltaic power forecasting model based on timegan, soft DTW-based K-medoids clustering, and a CNN-GRU hybrid neural network," *Energy Rep.*, vol. 8, pp. 10346–10362, 2022.
[51] Y. Liu, Q. Shi, Y. Wang, X. Zhao, S. Gao, and X. Huang, "An enhanced ensemble approach for non-intrusive energy use monitoring based on multidimensional heterogeneity," *Sensors*, vol. 21, no. 22, 2021, Art. no. 7750.

PASCAL A. SCHIRMER received the B.Eng. degree in electrical engineering from the University of Applied Sciences, Esslingen, Germany, in 2018, and the Ph.D. degree in electrical engineering from the University of Hertfordshire, Hatfield, U.K., in 2021. He is currently a Visiting Research Fellow with the University of Hertfordshire, where he focuses on NILM (software and hardware applications). He is also a Visiting Lecturer on electromobility with TAE, Esslingen. Since 2021, he has been with R&D Department BMW AG, Munich, Germany, where he is responsible for the lifetime evaluation of power electronic systems.

DIMITRIOS KOLOSOV received the Bachelor of Engineering (B.Eng.) degree in electronic and electrical engineering from Edinburgh Napier University, Edinburgh, U.K., in 2016, and the Master of Science (M.Sc.) degree in embedded intelligent systems from the University of Hertfordshire, Hatfield, U.K., in 2021. He is currently working toward the Ph.D. degree with the School of Physics, Engineering and Computer Science. His research interests include the development and acceleration of novel and innovative techniques for machine learning algorithms performing on various hardware platforms that are aimed for edge deployment, mainly vision, and energy disaggregation. He has a strong industry experience in FPGAs.

IOSIF MPORAS received the Dipl.-Eng. and Ph.D. degrees in electrical and computer engineering from the University of Patras, Patras, Greece, in 2004 and 2009, respectively. He is currently a Professor of signal processing and machine learning with the University of Hertfordshire, Hertfordshire, U.K. He has authored or coauthored more than 100 research publications cited more than 2,732 times (h-index: 27). His research interests include applications of signal processing and machine learning. He has participated in several U.K. and EU-funded research and development projects. He is a reviewer of grant applications, several international journals, and conferences.