Non-intrusive appliance load monitoring using low-resolution smart meter data

Jing Liao, Georgia Elafoudi, Lina Stankovic, Vladimir Stankovic
Department of Electronic and Electrical Engineering
University of Strathclyde, Glasgow, United Kingdom
Email:{jing.liao, georgia.elafoudi, lina.stankovic, vladimir.stankovic}@strath.ac.uk

***Abstract*—We propose two algorithms for power load disaggregation at low-sampling rates (greater than 1sec): a low-complexity, supervised approach based on Decision Trees and an unsupervised method based on Dynamic Time Warping. Both proposed algorithms share common pre-classification steps. We provide reproducible algorithmic description and benchmark the proposed methods with a state-of-the-art Hidden Markov Model (HMM)-based approach. Experimental results using three US and three UK households, show that both proposed methods outperform the HMM-based approach and are capable of disaggregating a range of domestic loads even when the training period is very short.**

I. Introduction

Disaggregating individual appliance usage from the total, aggregated energy consumption, is referred to as non-intrusive appliance load monitoring (NALM) [1]. Load disaggregation [1] is beneficial to customers to determine which appliances are the most energy consuming ones, which are faulty, and when it is time to retrofit old appliance; to suppliers to better forecast demand; to system operators to monitor the effect of smart grid fluctuations on the residential microgrid; to appliance manufactures and policy makers, and is an important aspect of smart homes/buildings and smart grids.

Since [1], many NALM algorithms have been proposed to adapt to advances in sensor technology capturing energy measurands at a range of sampling rates, generally in the order of kHz. However, more recently, as smart meter deployments are underway across the world and with increased awareness of domestic energy efficiency, NALM algorithmic work is focusing on cost-effective equipment that samples data at rates under 1Hz. It is not only the cost of the sensing technology, but also computational and storage cost as well as implementation efficiency that are key drivers towards the wide deployment of low-sampling smart meters [2]. However, so far, there are no widely available efficient solutions for NALM, with high accuracy, at such low sampling rates [3].

In [4], four NALM methods are proposed using (conditional) factorial hidden Markov model (FHMM) and Hidden semi-Markov models. Using active power measurements collected every 3sec together with duration and time of use of appliances, accuracy ranging from 72% to 99% was obtained for seven different houses with up to 10 appliances with an average accuracy of 83%. This method cannot disaggregate base load, such as, refrigerator, and is prone to converge to a local minimum. In [5] an FHMM is used for disaggregation of active power load. Although this is an unsupervised method, that uses expert knowledge to set initial models for states of known appliances, the models' operation for reliable results depends on correctly setting the a priori-values for each state for each appliance and using a training set where appliance operation does not overlap. A similar FHMM-based approach is tested in [6], where it is shown that the disaggregation accuracy drops for up to 25% when different houses are used to set the initial models compared to the case when the same house is used for building the models and testing. In [7], an unsupervised Additive Factorial Approximate Maximum A-Posteriori (AFMAP) inference algorithm is proposed using differential FHMMs. First, all snippets of active power data are extracted using a threshold and modelled by an HMM; next the k-nearest-neighbor graph is used to build nine motifs that are treated as HMMs over which AFMAP is run. The results show average accuracy of 87.2% using 7 appliances and sampling rate of 60Hz.

This paper focuses on NALM with sampling rates in the order of seconds and minutes, using active power measurements only and propose two approaches using *Dynamic Time Warping* (DTW) and a *Decision tree* (DT) algorithm. DTW is a time-series based approach that is ideal for comparing vectors of different lengths, with non-identical values. As with HMM, DTW is a very popular tool in speech recognition [8] and lately in data mining. It is used in [9] for water disaggregation together with HMM. The proposed DT-based approach is a low-complexity supervised approach, that can be trained using a very small aggregate dataset. It uses only active power changes during appliance state transitions to quickly identify appliances while ignoring power fluctuations within each state. This works well if the power range for each appliance is unique and there are no large variations of power within each appliance state. On the other hand, the proposed DTW-based method is unsupervised and scalable in the sense that it does not require re-training of the entire dataset when a new appliance is added. In contrast to DT, DTW exploits all collected power values to perform template matching. The two methods share the same pre-processing step to identify and remove corrupted data, and the same event detection algorithm; these significantly reduce the DTW library size and hence classification/pattern matching complexity. We provide reproducible algorithm description and validate the proposed methods using real-world measurements from three US houses from the REDD dataset [6] and three UK houses from the REFIT dataset [10] over a range of typical household appliances and compare our results to a state-of-the-art HMM-based approach [5].

In [11], the potential of using DTW for disaggregation

978-1-4799-4934-2/14/$31.00

is discussed, using energy consumption, rising edge count and Discrete Fourier Transform (DFT) as diagnostic features calculated over a window. It is shown that in contrast to speech applications of DTW, DFT is not needed, due to relatively small variations in the load within one window. The performance of the approach is unclear since only one-day's worth data is used for testing with sampling rate of 10sec in laboratory conditions. In [12], a DT classifier is used for pattern matching. However, in contrast to our work where we use active power samples at less than 1Hz, [12] uses transient features in addition to active and reactive power and collects samples of voltage and current at 100kHz. Using human-supplied labels corrected with the generalized likelihood ratio for event detection, the proposed DT algorithm achieved accuracy of only 64% compared to the 1-nearest neighbour classifier with the Fourier regression coefficients that achieved 79% when tested in laboratory conditions with eight loads.

We argue that in contrast to [9] and [11], HMM is unnecessary, because multi-state appliances have very characteristic signatures and hence can readily be detected via DTW, benefits from exploiting temporal correlation in running appliances are negligible due to stochastic human behaviour, and though HMM-based methods can be unsupervised [4], as our experiments show, they are sensitive to initialization and require relatively larger training sets than our proposed methods.

II. Proposed Disaggregation Algorithms

Due to the low energy monitoring sampling rate, the focus of the proposed methods is on steady state information. Both proposed algorithms use active power measurements, differ only in the way classification is performed, and comprise of two phases: a training phase and a testing phase. Training for both methods is done on aggregate data: (i) the DT-based method is a supervised approach as it also requires a labeled dataset for training, which is obtained from time-diaries or equivalent, (ii) the unsupervised DTW-based approach collects electrical power profiles, using only unlabelled aggregate data, to populate its library.

The entire disaggregation procedure can be summarised into the following four steps:

1) preprocessing: identify corrupted data and according to the error type, amend (or remove) the faulty data;
2) event detection: detect changes in time-series aggregate load curve due to one or more appliance runs over the base load;
3) feature extraction: once events are detected, the electrical features, such as edge range, profile between edges, duration, are isolated for each event or appliance run;
4) classification and pattern matching: classify (or match) the extracted features to the related electrical appliances.

Note that only the first three of the four above disaggregation steps are used for training, whereas all four steps are used for testing. In Step (4), the DT-based method disaggregates the load by classifying the extracted features using a DT model, while the DTW-based method calculates the best match between the extracted feature window and profiles in the library.

A. Problem Formulation and Notation

Let $\mathbf{M}$ be a set of all known household's appliances. Let $p(t_i)$ be the measured active power at time t_i. Without loss of generality, in the following we denote $p(t_i)$ as $p(t_i) = p(iT) = p(i)$, where $T = t_i - t_{i-1}$ is the sampling interval. We can express $p(i)$ as

$$p(i) = \sum_{j=1}^{M} p_j(i) + n(i), \tag{1}$$

where $p_j(i) \geq 0$ and $n(i)$ are the power load of Appliance j and measurement noise, respectively, at time instance iT. Note that $p_j(i)$ is zero if the appliance is off. Measurement noise comprises acquisition and communications noise as well as consumption of all unknown appliances. Given $p(i)$, the disaggregation task is to find $p_j(i)$, for all $j \in \mathbf{M}$ and all $i = 1, \ldots, N$.

For $i = 2, \ldots, N$, let $\Delta p(i) = p(i) - p(i-1)$. For an $n \times m$ matrix $\mathbf{A}$, $A(i,j)$ denotes matrix entry at row i and column j.

B. Event Detection

To achieve disaggregation, it is necessary to detect when an appliance is switched on or off. If $|p_j(i) - p_j(i-1)| \geq W$ we say that the appliance j has *changed state* at time instant iT, where W is an adaptive threshold. Threshold W depends on the appliance set $\mathbf{M}$ and must be set low enough so that for all j, if $|p_j(i) - p_j(i-1)| \leq W$ Appliance j did not change its state and, otherwise, it did. All appliances that operate below threshold W will not be detected. W is set based on the minimum state transition that needs to be detected and the maximum variation of the active power within one appliance state across all appliances' states; that is

$$W = \max\{\min_{m \in \mathbf{M}} \mathbf{p_m}, \max_{m \in \mathbf{M}} |\max(\mathbf{p_m}) - \min(\mathbf{p_m})|\}, \tag{2}$$

where $\mathbf{p_m}$ is a vector of readings of appliance m. Since the disaggregation process works iteratively by detecting one appliance i at a time and removing $\mathbf{p_i}$ from $\mathbf{p}$, W will be adaptively changed whenever one appliance is detected and removed.

An *event* occurs whenever an appliance changes its state. Edge detection is used to detect events by comparing $|p(i) - p(i-1)|$ with W. We say an event started at time l_s and ended at l_e if an appliance changed its state at l_s and l_e, and

$$|[p(l_s) - p(l_s - 1)] + [p(l_e) - p(l_e - 1)]| \leq C, \tag{3}$$

where C is a parameter, that is in practice estimated heuristically based on the detected rising edge. The higher the rising edge (device with high power consumption), the lower the value of C, e.g., C for electrical shower is 5% and C for refrigerator is 25%.

C. Feature Extraction

After detecting all events, it is necessary to extract features from each event. For the DT-based method, we extract the first increasing edge at the start of the event and the last decreasing edge at the end of the event, that is, $|\Delta p(l_s)|$ and $|\Delta p(l_e)|$, respectively. Thus, only two values, comprising the DT appliance signature, need to be stored for each detected event. On the other hand, for the DTW-based method all active power values, comprising the DTW appliance signature, between the first rising edge and the last decreasing edge at the end of the event are stored into a library.

D. Training and Pattern Matching

During the training phase with the DT-based method, all stored event features are examined along with the logs, that show which appliances triggered the event. All events that correspond to the same appliance are grouped, keeping only the maximum and minimum E(k,1) and E(k,2) for that appliance and discarding the rest. Thus, in total, only $2|\mathbf{M}|$ values are kept to design a DT model. This significantly reduces the memory and computational requirements. The DT is generated through the ID3 algorithm [13], where the selected attribute is the active power edge and the split criterion is based on minimum entropy [13].

Similarly, during the DTW-based training, a set of unique signatures is stored in the library. Whenever a new event is detected, it is checked against all signatures currently present in the library and if a match is found, the new signature is not included.

Comparison between different signatures is performed using DTW pattern matching explained next. Given two vectors of possibly different lengths, DTW performs non-linear mapping of one vector to another by minimizing the distance between them via dynamic programming. Let p and q be two extracted events of measurements of possibly different length. Let $d(x, y) = |x - y|$ for any two readings x and y. We initialize the distance measure $D(0, 0) = 0$ and $D(i, 0) = D(0, i) = \infty$, for all $i > 0$. Then, Algorithm 1 finds the distance between p and q. $D(i, j)$ is the accumulated DTW distance between points $p(1)$ to $p(i)$ and points $q(1)$ to $q(j)$, and $D(n, m)$ is the final distance between the two vectors.

Algorithm 1 DTW pattern matching: Find a distance between two vectors of possibly unequal lengths.

```
function DTW(p, q)
    n = length(p), m = length(q)
    for i = 1, ..., n do
        for j = 1, ..., m do
            d(i, j) = |p(i) − q(j)|
            D(i, j) = d(i, j) + min{D(i − 1, j), D(i − 1, j − 1),
                D(i, j − 1)}
        end for
    end for
    Output D(n, m)
end function
```

Let $\mathbf{DTWlib}$ be a library of signatures, where each row in $\mathbf{DTWlib}$ contains one unique signature. Let Q be the total number of rows (signatures) in $\mathbf{DTWlib}$. Using expert knowledge, each library signature is mapped to one appliance. A DTW threshold is used to remove similar signatures that come from the same appliance. It trades off performance and complexity: larger threshold would allow for more signatures (i.e., multiple signatures for the same appliance due to variations in operation) to be kept.

E. Classification and Testing

DT is organized as a tree of nodes, where each $node$ that is not a $leaf$, branches into two children nodes, $left_{child}$ and $right_{child}$, based on the split point value $V(node)$ [13]. Each *leaf node* is associated with an appliance or undefined value, that is, for each leaf $node$, Appliance$(node) \in \mathbf{M} \bigcup$ *Undefined value*. The classification process consists of going across the tree based on the split point value starting at the *root node* and ending in a leaf node. The ending leaf node will determine the classified appliance. If the classification terminates in a leaf node associated with an undefined value, then, classification error is declared. Algorithm 2 summarizes the classification procedure, where $\mathbf{DTmodel}$ is the trained DT model, and T is a rising edge in the new detected event.

Algorithm 2 DT classification: Classify the new event T using trained DT model.

```
function DTPREDICT(T, DTmodel)
    node = root
    repeat
        if T < V(node) then
            node = left_child
        else
            node = right_child
        end if
    until node is a leaf in DTmodel
    class=Appliance(node)
    return class
end function
```

For the DTW-based method, classification is performed via pattern matching. That is, each identified event is compared to each signature in the library using Algorithm 1, and the best match is found in terms of minimizing $D(n, m)$ distance measure over all library entries. Note that the output of the DTW classification is a soft value - $D(n, m)$ - the minimum distance between the testing event and the best matching signature from the library. The DTW classification algorithm is summarized in Algorithm 3, where the new event T is matched to the MIN-th signature in $\mathbf{DTWlib}$.

Algorithm 3 DTW classification: Compare the new event T with all signatures in the library.

```
function DTWCLASS(T,DTWlib)
    D = DTW(T, DTWlib(1, :)), MIN = 1
    for i = 2 : Q do
        if DTW(T, DTWlib(i, :)) ≤ D then
            MIN = i
        end if
    end for
    return MIN
end function
```

During the *testing phase*, the collected measurements are first fed to the edge detection algorithm which returns detected events that are then passed to Feature Extraction. For each extracted event, the DT-based method walks through the tree to find the optimal path. The DTW-based method compares the extracted events with library signatures.

Testing is performed iteratively, where in each iteration the threshold W is changed, i.e., at each iteration, one appliance, starting from the highest consumer, is detected and removed. The set of appliances is updated, and W is recalculated using (2). Decreasing W in each iteration facilitates detecting small

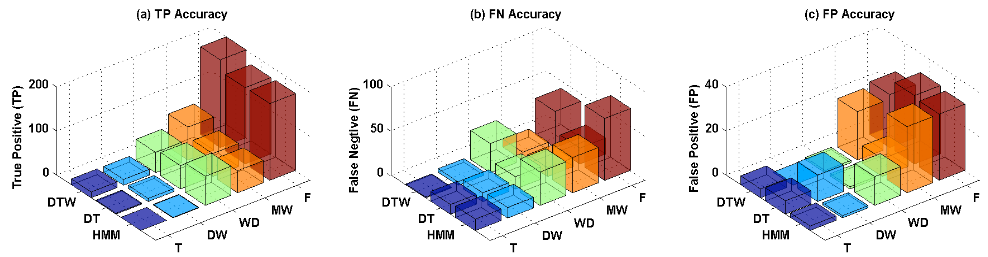

Fig. 1. TP, FN, and FP for the three methods using House 1 from REDD database. T is toaster, DW is dishwasher, WD is washer dryer, MW is microwave and F is refrigerator.

TABLE I. COMPARISON BETWEEN THE THREE METHODS USING REDD DATASET. ALL RESULTS ARE AVERAGED OVER ALL KNOWN APPLIANCES.

	PR(%)			*RE*(%)			F_M(%)		
House	DTW	DT	HMM	DTW	DT	HMM	DTW	DT	HMM
House 1	85.01	78.09	79.29	79.72	78.09	62.72	82.28	78.09	70.04
House 2	89.86	80.80	51.80	84.39	82.03	62.79	87.04	81.41	56.77
House 6	98.86	76.14	99.30	81.22	75.76	74.74	89.17	75.94	85.29

loads after high loads are detected and removed. This enables disaggregation even when multiple appliances are running at the same time, which was often the case in the used datasets.

III. EXPERIMENTAL RESULTS AND DISCUSSION

We evaluate the proposed disaggregation methods on six datasets, namely: (i) three US houses from the REDD database downsampled to 1min resolution, (ii) actual smart meter data from three UK houses at 6sec sampling rate from the REFIT database [10].

All households' appliances for which individual power consumption was not available, were considered unknown and hence they contribute to noise $n(i)$ in (1). This includes lighting in all six examples. We compare all our results with the state-of-the-art HMM-based method of [5] designed for low-sampling rates. For each dataset, all three tested algorithms always use the same amount of data for training and testing. The HMM-based method [5] requires prior initialization of the model using expert knowledge (state variances, mean value for each state and state transition probabilities), which was carried out in our experiments either using the information provided by the authors of [5], or were generated during training.

A. Evaluation Metrics

The evaluation metrics used are precision (PR), recall (RE) and F-Measure (F_M) [14] defined as: (i) $PR = TP/(TP + FP)$, (ii) $RE = TP/(TP + FN)$, and (iii) $F_M = 2 * (PR * RE)/(PR + RE)$, where TP, FN and FP, for each detected event are defined as:

- If the event is classified as m_j, and m_j was on, then the disaggregation is true positive (TP).
- If the event is classified as m_j, but m_j was off, then the disaggregation is false positive (FP).
- If the event is not classified as m_j, and m_j was on then the disaggregation is false negative (FN).

B. Results with US Houses

We use House 1, House 2 and House 6 in the REDD database. One week of data is used for training, i.e., 20% of the total dataset. Our results averaged over all known appliances, are shown in Table I. Table I shows that the proposed two methods in both House 1 and House 2 outperform HMM, and the DTW-based method is the best performing across all three houses. HMM has good performance in disaggregating refrigerator events, thus it overall performs well in House 6, where refrigerator events are very frequent compared to the events of the other appliances. Note that refrigerator is always on, hence there are more samples to train than for other appliances, leading to improved HMM performance. On the other hand, HMM performs poorly for dishwasher and similar appliances that run only occasionally, and require large training dataset to generate a good Markov model.

Our per appliance results for House 1 are shown in Fig. 1. It can be seen from the figure that both proposed DT-based and DTW-based methods outperform the HMM-based method for all five appliances. Indeed, as shown in Fig. 1, our proposed methods have a higher TP rate than HMM for all appliances and generally lower FN rate. The DTW-based method shows the best performance overall and for most individual appliances. The HMM-based method failed to correctly identify microwave, dishwasher and toaster in most of the cases. The DT-based method has also poor performance for the toaster and dishwasher, due to the fact that the toaster occurrence was often confused with other similar unknown appliances present in the house.

C. Results with UK Houses

We use active aggregate power measurements, sampled at 6sec, from [10]. Two months of data are used: one week for training and the rest for testing.

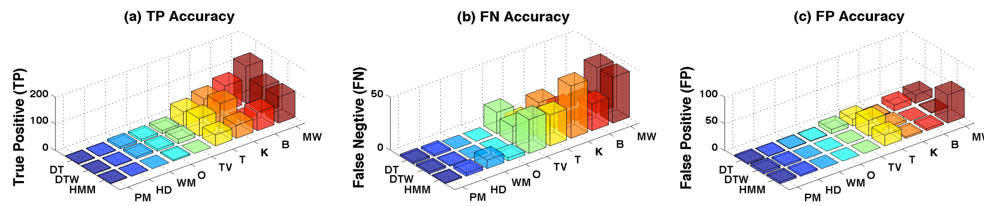

Fig. 2. TP, FN, and FP values for the three NALM methods for UK test House 1. PM is pancake maker, HD is hair dryer, WM is washing machine, O is oven, TV is television, T is toaster, K is kettle, B is boiler and MW is microwave.

TABLE II. COMPARISON BETWEEN THE THREE METHODS FOR THE THREE UK TEST HOUSES. ALL RESULTS ARE AVERAGED OVER ALL KNOWN APPLIANCES.

	PR(%)			*RE*(%)			F_M(%)		
House	DTW	DT	HMM	DTW	DT	HMM	DTW	DT	HMM
House 1	88.32	87.54	75.35	80.60	92.35	57.95	84.28	89.88	65.52
House 2	68.45	95.81	32.90	78.86	78.81	78.28	73.29	86.49	46.33
House 3	69.53	87.26	30.25	75.55	99.08	83.23	72.42	92.79	44.37

Our results for all three UK test houses, averaged over all known appliances, are shown in Table II. It can be seen that the proposed DT-based method has consistently high performance across all three houses. DTW-based method shows a lower accuracy for Houses 2 and 3, because there were more unknown appliances in these two houses. Overall, the two proposed methods again outperform HMM.

Table III shows disaggregation results for the microwave for all 3 houses. One can see that the large number of FPs reduced the overall performance of all algorithms in Houses 2 and 3. A drop in performance for House 3 is evident, due to the fact that many unknown appliances were confused with the microwave. The poor performance for microwave detection accuracy reduces the overall accuracy, as seen from Table II.

Disaggregation results per appliance for House 1 are shown in Figure 2. It is interesting that the HMM-based approach did not recognize any of 8 runs of washing machine, probably because there were many zeros during the operation, which confused state-detection. The HMM-based method is also unsuccessful with recognizing TV. This is probably due to the fact that TV is usually on for a long time, thus there was no instance in the training set when TV was running alone, with no other appliance running in parallel. Other problems associated with detecting TV activity are that TV power values vary widely, as well as the duration of TV runtime from one day to the next. On the other hand, the proposed approaches can successfully be trained even when appliance usage overlaps.

The DTW-based method identified all instances of oven and washing machine. The proposed methods detect hair dryer events at higher detection accuracy compared to HMM although all three tested NALM methods for this particular appliance have a relatively lower detection accuracy than for other appliances; this is due to the fact that the hair dryer was often confused for a rice cooker, which is considered an unknown appliance since it does not occur in the training data. Rice cooker and hair dryer have similar signatures. Overall, the DT-based method gives the best results, which is mainly due to poor DTW performance for the boiler, which is a very low consuming appliance.

TABLE III. COMPARISON OF MICROWAVE DISAGGREGATION RESULTS BETWEEN THE THREE METHODS FOR THE THREE UK TEST HOUSES.

House	*Method*	*TP*	*FP*	*FN*	*PR*	*RE*	F_M
House 1	*DTW*	98	2	43	98	65.5	81.33
	DT	139	16	2	89.68	98.58	93.92
	HMM	98	53	43	64.9	69.5	67.12
House 2	*DTW*	269	297	84	47.53	76.2	58.54
	DT	260	0	93	100	73.65	84.83
	HMM	267	543	86	32.96	75.64	45.91
House 3	*DTW*	367	222	131	62.31	73.69	67.53
	DT	498	385	0	56.4	100	72.12
	HMM	400	1428	98	21.88	80.32	34.39

D. Summary

The two proposed approaches share the same preprocessing and event detection steps, thus the performance difference can be attributed to different classification steps. The proposed DT-based method is a low-complexity approach that requires the least storage and computational resources. It operates purely on the rising and falling active power edges at the start and end of events, and requires only a small sample of training data to build a DT model. It is a supervised approach since it is trained on labeled aggregate data. The proposed DTW-based method needs a library of appliance signature, hence requires more storage space; it is computationally more expensive than the DT-based method as it performs pattern matching between the extracted window and all signatures present in the library during the classification step. The complexity is thus driven by the number of signatures stored in the library. In the above experiments, for most appliances only one signature is kept, and there was no appliance with more than three signatures kept in the library after training. Since training is done on an unlabeled aggregate data, it is an unsupervised approach. It is also scalable, since in contrast to HMM and DT, it is not retrained when a new appliance is added. For any appliance, DTW-based method performs pattern matching using the library signatures, whereas DT-based model and HMM methods classify the appliances according to the trained models.

Experiments demonstrate that both proposed methods are successful for very small training datasets. HMM-based approaches require state variances and probabilities of state transitions which are difficult to estimate using a small training set; moreover, unlike the proposed methods, they usually require appliances running alone during training for successful modelling.

The DTW-based approach performs better with the lower resolution dataset (1min) than the DT-based method because it is very sensitive to fluctuations present in the 6sec dataset. The DTW-based method has difficulty recognizing low load appliances, such as boiler, but its strength lies in detecting multi-state appliances such as oven and washing machine. The DT-based approach performs very well for the appliances whose operation time significantly varies and tends to be long, such as TV, which is not the case with the tested HMM-based approach. Interestingly, the DTW-based approach shows good performance for the TV demonstrating that it is not sensitive to fluctuations in event duration. The HMM-based approach performs well for the refrigerator, since many instances are available for training, and this appliance is usually the only one running at night, facilitating good modelling. All approaches were able to identify the highest consuming appliances, e.g. washer-dryer, which would provide useful information to customers and possibly a more efficient usage plan of these appliances.

IV. Conclusion

We proposed two solutions for power load disaggregation using active power at low sampling rates: a low-complexity, supervised, DT-based method and an unsupervised DTW-based approach. Experimental results using three US and three UK houses show the competitiveness of the proposed solutions where the proposed algorithms demonstrate consistently improved detection performance for a range of household appliances, with respect to a state-of-the-art HMM-based approach. Indeed, the proposed methods showed a success rate in the range of 85-90% in the test house with nine appliances as opposed to 66% obtained with the HMM-based approach. Moreover, while the proposed DT-based approach records only the active power range per appliance during training and DTW-based approach only time-series active power values of the appliance, the HMM-based approach needs more detailed statistics, such as state transition probabilities that usually require a larger dataset for training. This is the key reason for the success of the proposed approaches when dealing with limited training sets and presence of many unknown appliances.

In future work: (i) robustness of the two algorithms needs be tested with respect to missing data, limited training set, and the case when only manufacturer's manual's are used; (ii) exploitation of DT, for generating the training library, and DTW, for pattern matching, will be investigated; (iii) implementation of DTW for performing online training during (instead of prior to) disaggregation will strengthen scalability; (iv) incorporation of other sensors, such as temperature, light intensity, to improve disaggregation performance; (v) testing the proposed approaches for disaggregating gas and water consumption.

Acknowledgment

This work is supported in part by the UK Engineering and Physical Sciences Research Council (EPSRC) projects REFIT EP/K002708, under the Transforming Energy Demand in Buildings through Digital Innovation (BuildTEDDI) funding programme. The authors would like to thank O. Parson for sharing his HMM code.

References

[1] G. W. Hart, "Non-intrusive appliance load monitoring", *Proc. of the IEEE,* vol. 80, no. 12, pp. 1870–1891, Dec. 1992.

[2] K.C. Armel, A. Gupta, G. Shrimali, and A. Albert, "Is disaggregation the holy grail of energy efficiency? The case of electricity," *Energy Policy*, vol. 52, pp. 213–234, 2013.

[3] A. Zoha, A. Gluhak, M.A. Imran, and S. Rajasegarar, "Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey," *Sensors,* vol. 12, pp. 16838–16866, Dec. 2012.

[4] H. Kim, M. Marwah, M. Arlitt, G. Lyon, and J. Han, "Unsupervised disaggregation of low frequency power measurements," in *Proc. 11th SIAM Int. Conf. Data Mining,* Mesa, AZ, April 2011.

[5] O. Parson, S. Ghosh, M. Weal, and A. Rogers, "Non-intrusive load monitoring using prior models of general appliance types," in *Proc. the 26th Conf. Artificial Intelligence (AAAI-12)*, Toronto, CA, pp. 356–362, July 2012.

[6] J. Kolter, and M. Johnson, "REDD: A public data set for energy disaggregation research," in *Workshop on Data Mining Applications in Sustainability (SIGKDD)*, San Diego, CA, 2011.

[7] J. Kolter, and T. Jaakkola, "Approximate inference in additive factorial HMMs with application to energy disaggregation," in *J. Machine Learning*, vol. 22, pp. 1472–1482, 2012.

[8] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoustics, Speech, and Sig. Proc*., vol.26, no.1, pp. 43–49, 1978.

[9] K. A. Nguyen, H Zhang, and R. A. Stewart, "Development of an intelligent model to categorise residential water end use events," *J. Hydro-environment Research*, March 2013.

[10] REFIT: Personalised Retrofit Decision Support Tools for UK Homes using Smart Home Technology, http://www.refitsmarthomes.org/index.php/refit-launches.

[11] A.A. Zaidi, F. Kupzog, T. Zia, and P. Palensky, " Load recognition for automated demand response in microgrids," *In Proc. IECON-2010 36th Annual Conf. IEEE Industrial Electronics Society,* Phoenix, AZ, Nov. 2010

[12] M. Berges, E. Goldman, H. S. Matthews, and L. Soibelman, "Learning systems for electric consumption of buildings," in *Proc. 2009 ASCE Int. Workshop Computing in Civil Engineering*, Austin, TX, 2009.

[13] J.R. Quinlan,"Induction of decision tree," *Machine Learning*, vol.1, no.1, pp.81-106, 1986.

[14] D.L. Olson and D. Delen, *Advanced Data Mining Techniques*, Springer, 2008.