Neural Computing and Applications
https://doi.org/10.1007/s00521-020-04916-5

S.I.: EMERGING APPLICATIONS OF DEEP LEARNING AND SPIKING ANN

On time series representations for multi-label NILM

Christoforos Nalmpantis[1] · Dimitris Vrakas[1]

Received: 18 November 2019 / Accepted: 6 April 2020

Abstract

Given only the main power consumption of a household, a non-intrusive load monitoring (NILM) system identifies which appliances are operating. With the rise of Internet of things, running energy disaggregation models on the edge is more and more essential for privacy concerns and economic reasons. However, current NILM solutions use data-hungry deep learning models that can recognize only one device and are impossible to run on a device with limited resources. This research investigates in-depth multi-label NILM systems and suggests a novel framework which enables a cost-effective solution. It can be deployed on an embedded device, and thus, privacy can be preserved. The proposed system leverages dimensionality reduction using Signal2Vec, is evaluated on two popular public datasets and outperforms another state-of-the-art multi-label NILM system.

Keywords Non-intrusive load monitoring · Multi-label NILM · Smart grids · Time series representation · Signal2Vec

1 Introduction

Efficient and accurate energy monitoring of domestic buildings can significantly diminish energy wastage. In the USA, the savings of the electricity consumption by residential buildings are estimated around 15 % of the total consumption or 200 billion kWh per annum. This amount is equivalent to 81.3 million tons of coal [13]. The advantages of energy monitoring are not restricted to the positive effect on the environment. It will affect many sectors in the industry by reforming building operations, advancing smart grid optimization, boosting energy consumption forecasting, etc. [2].

Non-intrusive load monitoring (NILM) has been a subject of research since 1992 [15]. It has been favored over intrusive methods due to economic and practical reasons. In this context, a NILM system needs to meet the requirements of a large-scale deployment in the real world. As far as hardware is concerned, most NILM systems are applicable because they offer lower costs and simpler installation. On the software side though, power disaggregation models can be computationally intensive and thus not financially viable. This is not a surprise as power disaggregation has been classified as NP-complete [15].

Numerous NILM approaches have been proposed, with the initial studies being based on combinatorial optimization techniques [3, 30, 50]. The main concept in these studies was to use a database of load signatures and solve the power disaggregation problem using combinatorial optimization algorithms. Probabilistic models have also been employed, where the states of the appliances' signals were identified based on hidden Markov models [35, 58, 59]. Later research focused on machine learning, including among others shallow neural networks, time series classification and clustering methods [17, 26, 44, 49]. Other methodologies tried to focus on feature extraction in combination with classifiers such as support vector machines or using extra information like the weather or the number of people in a house [10, 25, 30].

Most recent approaches include variations of hidden Markov models (HMMs) and deep neural networks. HMMs were the state of the art for long time, but their main drawback is that they are computationally intractable and do not scale for large number of appliances.

✉ Christoforos Nalmpantis
christofn@csd.auth.gr

Dimitris Vrakas
dvrakas@csd.auth.gr

[1] School of Informatics, Aristotle University of Thessaloniki, Thessaloníki, Greece

Published online: 02 May 2020

Deep neural networks have only recently been proposed and have demonstrated very promising results [38]. The most popular deep learning architectures are recurrent neural networks (RNNs) [18] and convolutional neural networks (CNNs) [56]. Although comparing different NILM solutions has been a non-trivial task [38], recent benchmarks have shown that deep learning methods clearly demonstrate state-of-the-art performance [9]. According to the benchmarks, there is no clear winner because the results also depend on the actual appliance which is disaggregated. The top four models are a deep neural network with five convolutional layers, named sequence-to-point (Seq2point), a similar architecture named sequence-to-sequence (Seq2seq), a recurrent neural network (RNN) and a distilled version of that named online GRU. A more exhaustive evaluation of these models, including a bigger variety of appliances, is advised to be conducted in the future.

Despite the high accuracy results, the deployment of a NILM system based on deep neural networks would require huge computing power. The root cause is manifold. Firstly, deep neural nets consist of millions of learning parameters, which make the phases of learning and inference heavy. Secondly, energy data are generated every second or minute, contributing to a vast amount of data. Thirdly, according to the majority of the proposed solutions, identifying an appliance requires a model dedicated to this appliance. Consequently, even a distilled version of a deep neural network, like the one named “online GRU” [24], would require a one-to-one model-appliance relationship.

The aforementioned problems make running NILM models on embedded devices prohibitive, whereas a cloud solution is very expensive. A viable solution should have the following properties: (a) a data representation which achieves sufficient dimensionality reduction, (b) a lightweight disaggregation model and (c) a one-to-many relationship, where one model can identify many appliances. The latter property can be addressed by handling NILM as a multi-label classification problem. The other two properties can be met by evaluating various lightweight multi-label machine learning models in combination with time series approximation methods.

In this paper, we focus on the above challenges and present a novel framework, called multi-NILM. It leverages time series representations to account for resource limitations when disaggregating the energy consumption of a building. Given this framework, we employ Signal2Vec, a recent algorithm that was firstly proposed in our previous short paper [39]. This work includes a systematic analysis of the model and a formal introduction of the average vector representation ($S2V_{avg}$) of a time series. Furthermore, there is a methodical comparison of the performance of Signal2Vec against the most popular existing time series representations for the task of energy disaggregation. To the best of the authors’ knowledge, this is the first time these algorithms are used in the problem of multi-label NILM.

This paper contributes to the problem of multi-label NILM by: (a) designing a novel framework, named multi-NILM, that tackles the NILM problem as a multi-label classification problem, (b) performing a systematic evaluation of eight popular time series representations in the context of multi-NILM and (c) introducing an integrated system for multi-label NILM based on Signal2Vec algorithm and multi-NILM framework, which outperforms one of the most reliable state-of-the-art systems.

The remainder of this article is organized as follows: Firstly, related work around multi-label solutions is revisited. Following, our novel framework is presented, explaining which problems it solves. Next, there is a short description of seven existing time series representations and a comprehensive analysis of Signal2Vec. Experiments are conducted including a systematic comparison of the various time series representations using the proposed multi-NILM framework. In the last section, there is an in-depth evaluation of our best model against an existing multi-label NILM proposal. Finally, there is a summary and suggestions for future work.

2 Background work

The goal of a NILM system is to break the total power consumption of a residence, into estimates of the actual power demand of each appliance. It is a single-source separation problem, and every sample is a mixture of signals that are not mutually exclusive. Considering now each target appliance as a binary label, the problem of power disaggregation can be seen as a multi-label classification task.

Multi-label classification was firstly introduced in NILM by Basu et al. [4, 5]. The proposed method used the aggregate power consumption at 10 min interval and identified three energy-hungry appliances. The algorithms under examination were: decision trees [53], boosting [55], RAkEL [53], MLkNN [57] and classifier chains [41]. During testing phase, the models showed encouraging results as far as the test period was no more than 6 months. For longer period, the performance dropped significantly. The authors attributed the performance degradation to the seasonal variations of pattern usage of appliances. Thus, a model in the real world would have to be retrained at least every 6 months. A similar approach was used in [6, 7], where multi-label classification models were compared against a standard single-label hidden Markov model.

Li et al. [29] proposed a different approach for multi-label NILM, using an expectation maximization semi-supervised method and converting energy time series to delay embeddings. The experiments used REDD dataset [23], where the sampling rate is 1 Hz. The usage of delay embedding via Takens' theorem [52] has been very successful and has been incorporated in more multi-label NILM systems [27]. Tabatabaei et al. [51] compared delay embedding and wavelet features in combination with RAkEL and MLkNN. The experiments disaggregated 6–7 appliances and MLkNN in time domain demonstrated the best results. The main database was REDD again with sampling rate 1 Hz, and the results varied from 0.39 to 0.61 macro-f1 score depending on the experiment.

The work of Tabatabaei et al. has been one of the best performing multi-label NILM solutions and has inspired the scientific community to discover more sophisticated multi-label NILM methodologies. Vermal et al. [54] proposed a model called multi-label restricted Boltzmann machine (ML-RBM), which surpassed Tabatabaei's et al. models in their experimental setup. This setup was restricted to 1 h time period per sample and the target included up to four appliances. In the same fashion, Singhal et al. [48] compared two techniques called deep dictionary learning and deep transform learning against Tabatabaei's et al. multi-label classifiers. The experiments involved input data equivalent to 1 h duration and target data with four appliances. Li et al. [28] suggested three new graph-based semi-supervised multi-label (SSML) algorithms, outperforming Tabatabaei's et al. models. The downside of the SSML algorithms is that they require quadratic time. According to the authors even downsampling the data, the duration of the experiments was very long. Loukas et al. [33] constructed a new dataset with sampling rate at 50 kHz and verified that the multi-label NILM performance is enhanced when higher harmonic currents are included as features. The best model was a decision tree showcasing promising results on the custom experimental dataset. Kim et al. [20] borrowed audio signal processing techniques and combined spectrogram and mel-frequency cepstral coefficients (MFCC) as inputs to a multilayer Long-Short-Term-Memory (LSTM) neural network. The suggested model was evaluated using accuracy and micro-f1 score, showing similar or better results when compared to other models. However, according to the literature these metrics are not the best ones for multi-label tasks and a recommended option would be macro-f1 score [53]. In short, the literature pertaining to the reproducibility and comparability of NILM systems strongly suggests that there should be a standardization and consensus on evaluation procedures [21, 38].

Since NILM has not been fully explored as a multi-label problem, there is plenty room of improvement. Two key considerations in this research are the amount of data and the hardware that will host the models in the real world. Speaking of that, new proposals should focus on dimensionality reduction and develop models with low computing demands in order to cut costs to minimum and enable privacy preservation.

3 Multi-NILM framework

In the era of Internet of things, there is an explosion of data with more than 50 billion IoT devices connected [42]. In the scope of non-intrusive load monitoring, smart electricity readers are producing a huge number of records per day. Electricity consumption of millions of buildings is logged at least hourly, sometimes with sampling rate up to 250 kHz [22]. Handling huge amount of data and extracting useful information using data mining techniques are indispensable and very challenging. Thereupon, it is crucial to develop an efficient framework which intrinsically exploits dimensionality reduction methods.

A novel framework, named multi-NILM, is proposed encapsulating three inherent properties that make it suitable for a real-world application. Figure 1 depicts multi-NILM and summarizes its properties. The first property utilizes a data representation for sufficient dimensionality reduction. Hence, less computing power is demanded for processing and the complexity of the data mining algorithms is decreased. The process of shortening the data can take place either on a cost-efficient cloud solution or on a local device with limited resources such as an embedded system, a smart phone and others. The local configuration enables privacy preservation, since there is no need to communicate data to the supplier. For the purpose of backup, data could be send encrypted. The second property of the framework is that it uses lightweight disaggregation models, making local inference possible and enhancing the benefits in terms of costs and privacy impact. The low complexity of the model stems from the low dimensional input. The last property is that it tackles the disaggregation problem as a multi-label classification problem, which means that just one model can identify many appliances. The benefits of a multi-label approach are described along these lines. Using one model promises lower computational complexity. On the contrary, running one model for each appliance can be much more expensive. A multi-label model is scalable because it can be retrained for each new device, whereas adding one extra model for a new device requires additional computational power. Another drawback of single-label NILM models is that the results of two or more models might be conflicting.

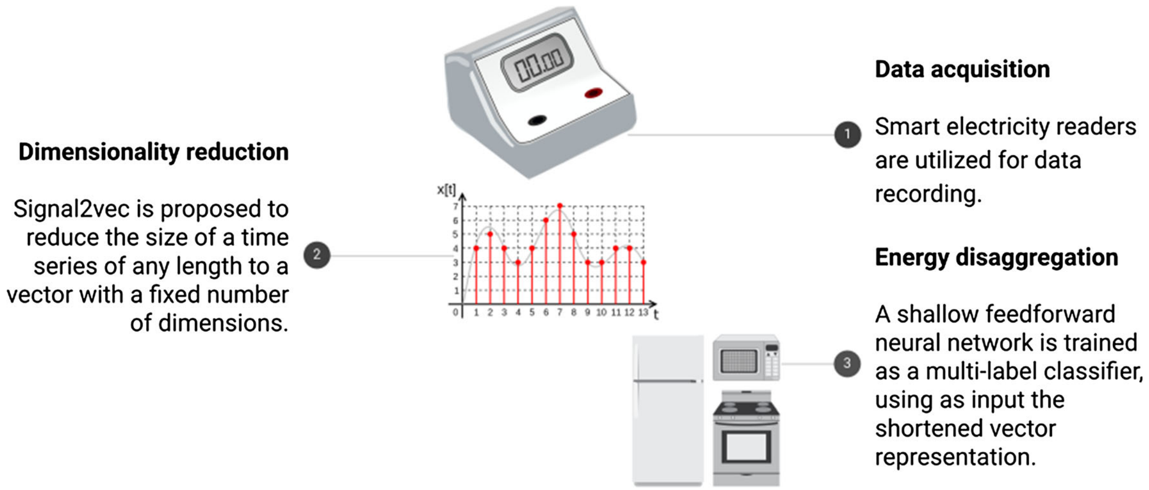
Data acquisition
1
Smart electricity readers are utilized for data recording.
Dimensionality reduction
Signal2vec is proposed to reduce the size of a time series of any length to a vector with a fixed number of dimensions.
2
x[t]
t
Energy disaggregation
3
A shallow feedforward neural network is trained as a multi-label classifier, using as input the shortened vector representation.

Fig. 1 An efficient framework for energy disaggregation

Multi-NILM framework is applied on the task of energy disaggregation but is general and can be applied on any other time series problem. It can also adopt any signal approximation algorithm in combination with a multi-label machine learning model. In our implementation, Signal2Vec is selected after a systematic comparison with the following time series representations: Piecewise Aggregate Approximation (PAA), Symbolic Aggregate Approximation (SAX), 1d-SAX, Discrete Fourier Transform (DFT), Symbolic Fourier Approximation (SFA), Bag-of-SFA-Symbols (BOSS) and Word ExtrAction for time SEries cLassification (WEASEL). On the side of the multi-label classifier, the current configuration uses a shallow feed-forward neural network, which was selected through a series of comparative experiments. The process of the configuration of the framework, the details of the time series representations and the classifier are described in the following sections.

4 Time series representations

For the purpose of configuring multi-NILM framework, seven of the most popular time series representations are selected. A brief description of these algorithms is presented below, outlining their advantages and disadvantages.

Piecewise Aggregate Approximation (PAA) One of the most popular time series approximations is Piecewise Aggregate Approximation (PAA) [19]. The implementation is very straightforward. Firstly, the series is divided into M time frames of equal size. Then, the mean value of each time frame is calculated, forming a representation of M dimensions. The algorithm of PAA is expressed as follows: Denote a time series $X = x_1, \ldots, x_n$, with length n. PAA is an approximation that represents time series X in M space by a vector $\bar{X} = (\bar{x}_1, \ldots, \bar{x}_M)$, where $M \leq n$. The *ith* element of $\bar{X}$ is calculated by the following equation:

$$\bar{x}_i = \frac{M}{n} \sum_{j=n/M(i-1)+1}^{(n/M)i} x_j \qquad (1)$$

It is proved that the complexity of this transformation can be reduced from O(nM) to O(Mm) where m is the number of frames. Figure 2 shows a diagram of a random time series including its PAA representation. From this diagram, it can be concluded that PAA transforms a given time series to a sequence of steps. As a result, this technique is proved to be quite robust when dealing with noisy data.

Symbolic Aggregate Approximation (SAX) SAX is an extension of PAA, inheriting its simplicity and the low computational complexity [31]. The main advantage of this algorithm is that it transforms a time series into a string representation. SAX has been very successful in many tasks related to time series such as classification, clustering, summarization, anomaly detection and indexing. The steps of the algorithms are: (a) Apply PAA algorithm to segment the time series. (b) Discretize the average values by mapping them to letters of an alphabet. The only requirement of the latter step is that the new symbols have to be produced with equiprobability. This is achieved with normalized time series because they have a Gaussian distribution. The difference between SAX and PAA is shown in Fig. 2. More formally, this is expressed by defining breakpoints as a list of ordered numbers $B = \beta_1, \beta_2, \ldots, \beta_{a-1}$ such that the area under the Gaussian curve between two adjacent breakpoints is constant. Consequently, a time series can be converted into words by assigning a symbol $alpha_j$ for each interval $[\beta_{j-1}, \beta_j)$. The mapping of PAA approximation $\bar{C}$ into a string $\hat{C}$ is given by the formula below:

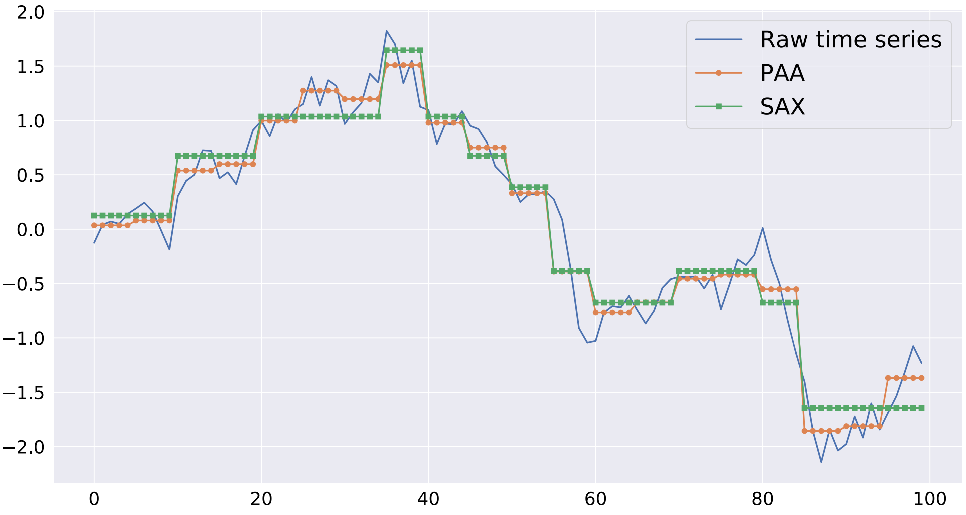

Fig. 2 Illustration of a random time series and its PAA and SAX representations

$$\hat{c} * i = alpha * j, \; iif \; \bar{c} * i \in [\beta_{j-1}, \beta_j) \quad (2)$$

1d-SAX An alternative version of SAX is 1d-SAX [34]. The benefit of this version is that it takes into account the trend of each subsequence when mapping the values of the time series into symbols. Thus after applying PAA, instead of computing the average of a segment, we compute the linear regression. The two coefficients of linear regression for each segment are mapped into symbols independently and then combined together into one symbol. The statistical properties of the average values and the slope values form a Gaussian distribution, and consequently, the quantization step is achievable in the same way as in the original version of SAX.

Discrete Fourier Transform (DFT) One of the most common time series approximation algorithms is the Discrete Fourier Transform (DFT), which was firstly used for the problem of time series similarity search by Agrawal et al. [1]. The most important properties of DFT are that its coefficients consist an orthogonal basis, the first few coefficients contain most of the information, it is fast to compute due to Fast Fourier Transform (FFT) algorithm and it preserves the Euclidean distance due to Parseval's theorem.

Symbolic Fourier Approximation (SFA) Schäfer et al. [46], inspired by DFT, suggested a novel time series representation named Symbolic Fourier Approximation. SFA consists of two steps: preprocessing and transformation. Preprocessing includes the DFT approximation and a quantization technique called multiple coefficient binning (MCB). MCB minimizes information loss during quantization by grouping the DFT coefficients of all subsequences, creating a histogram for each group and applying binning to each group. The transformation step maps the MCB discretization into symbols of a finite alphabet.

Bag-of-SFA-Symbols (BOSS) An extension and improvement of SFA is a time series representation named Bag-of-SFA-Symbols (BOSS), which is very robust in noise [45]. This feature is important in time series tasks, as real-world data tend to be very noisy or erroneous. In order to obtain BOSS representation of a given time series T, sliding windows $S_{i;w}$ of size w are extracted. Next, each window is converted to unordered SFA words. The transformations $SFA(S_{i;w}) \in \Sigma^l, for \; i = 1, 2, \ldots, (n - w + 1)$ are used to build a histogram. The BOSS histogram is a function, $B : \Sigma^l \rightarrow N$, which maps the SFA word space into the space of natural numbers. BOSS has the property of phase shift invariance because it does not take into account the order of SFA words and uses numerosity reduction [31, 32] to avoid outweighing segments with constant values. An example of BOSS transformation for a time series with two classes is shown in Fig. 3. It illustrates the words that are extracted from the given time series and the features representing frequencies of each word.

Word ExtrAction for time SEries cLassification (WEASEL) WEASEL is a novel time series representation which has demonstrated very promising results on the task of time series classification [47]. The efficiency of this method is attributed to two new methods named discriminative approximation and discriminative quantization. The former one uses the one-way ANOVA F test during the approximation step, to select the Fourier coefficients that are characteristic to class labels. The latter method maximizes the information gain, resulting in low entropy feature set of class labels. Given a time series, WEASEL firstly extracts normalized windows of various sizes. Then, Fourier coefficients are calculated and filtered, keeping the ones that are characteristic to this particular time series. The filtering process is achieved by using the ANOVA F-test. Next, the Fourier coefficients are quantized. The quantization

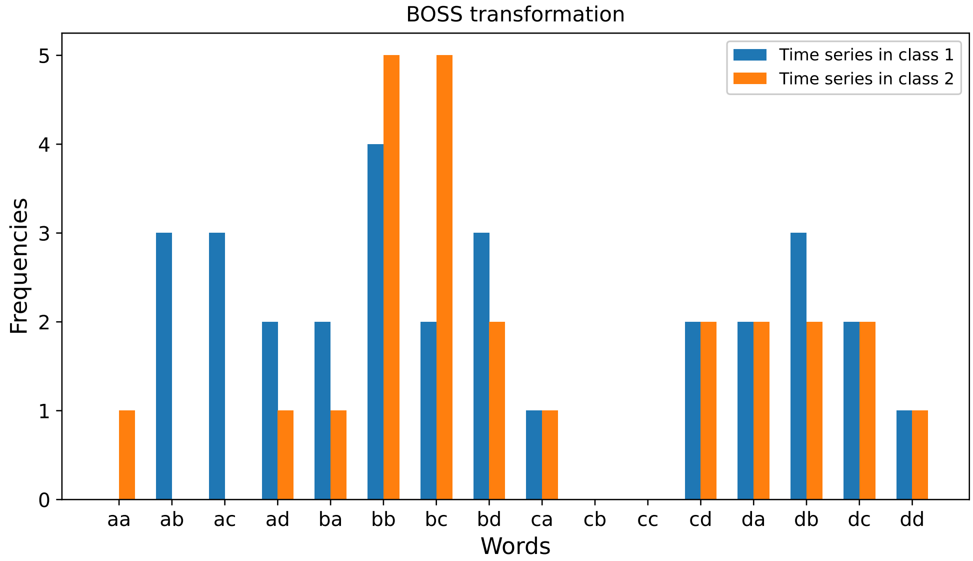

Fig. 3 Illustration of the words and their frequencies that have been learned by BOSS

process is completed using information gain binning to best separate the various time series classes. The unigrams and bigrams are then used to construct a bag-of-patterns, filtering out irrelevant words using the Chi-squared test. Bag-of-patterns approaches have linear computational complexity; therefore, WEASEL achieves the best trade-off between accuracy and speed.

5 Signal2Vec

Signal2Vec [39] is a novel algorithm that maps any time series into a vector space. It is a generalization of word2vec [36] and extends its applicability to sequences in continuous space. The two key concepts of Signal2Vec include a quantization process and the skip-gram model [36]. The latter is responsible for constructing the embedding feature space. The learning process of the embedding space is unsupervised, computationally efficient and scalable. The main advantage in comparison with other time series representation algorithms is that the embeddings are built only once and can be reused by other machine learning systems that are applied in unseen datasets.

An illustration of the entire workflow of Signal2Vec is presented in Fig. 4. The given time series is quantized via clustering, and each cluster defines a token. The number of clusters is estimated using silhouette score, which tells which objects fit well in their cluster [43]. The search space of the number of clusters is also constrained by a minimum and a maximum desirable number of clusters. In the domain of energy buildings, the hypothetical bounds of the space can be estimated by calculating the number of possible energy states using the complexity of power draws [12]. Once the clusters have been found, a function is trained to map any time series to a discrete sequence of tokens. This function can be a classifier, like k-nearest neighbors, which is trained only once and then can be used to map any new time series to the finite space of tokens.

In the terminology of word2vec, a token is a word, the finite space of tokens is called vocabulary and a sequence of tokens is called corpus. In order to apply the skip-gram model, we need to define a context.

Definition 1 *Context*: Given a sequence of tokens and a sliding window of length W, with W odd, let TKN_{target} be the middle token of the window. Then, the context consists of the rest of the tokens inside the window.

The objective of the skip-gram model is to predict the context given a specific token. The architecture is a shallow neural network with one hidden layer and is trained with pairs of tokens. One token is the target, and the other one belongs to the context. The network trains the weights of the hidden layer from the frequencies each pairing shows up. The formal definition of the objective function is given below.

Let $TKN_1, TKN_2, \ldots, TKN_n$ be a sequence of N training tokens. Then, the objective function tries to maximize the average log probability according to the formula:

$$1/N \sum_{t=1}^{N} \sum_{-c \leq i \leq c, i \neq 0} \log p(TKN_{t+i} | TKN_t), \tag{3}$$

where c is the training context.

The loss function of the neural network is noise-contrastive estimation (NCE) [14]. Adagrad [11] is the selected gradient-based optimizer, with learning rate 0.001. The size of each embedding is 300, and the context is six tokens.

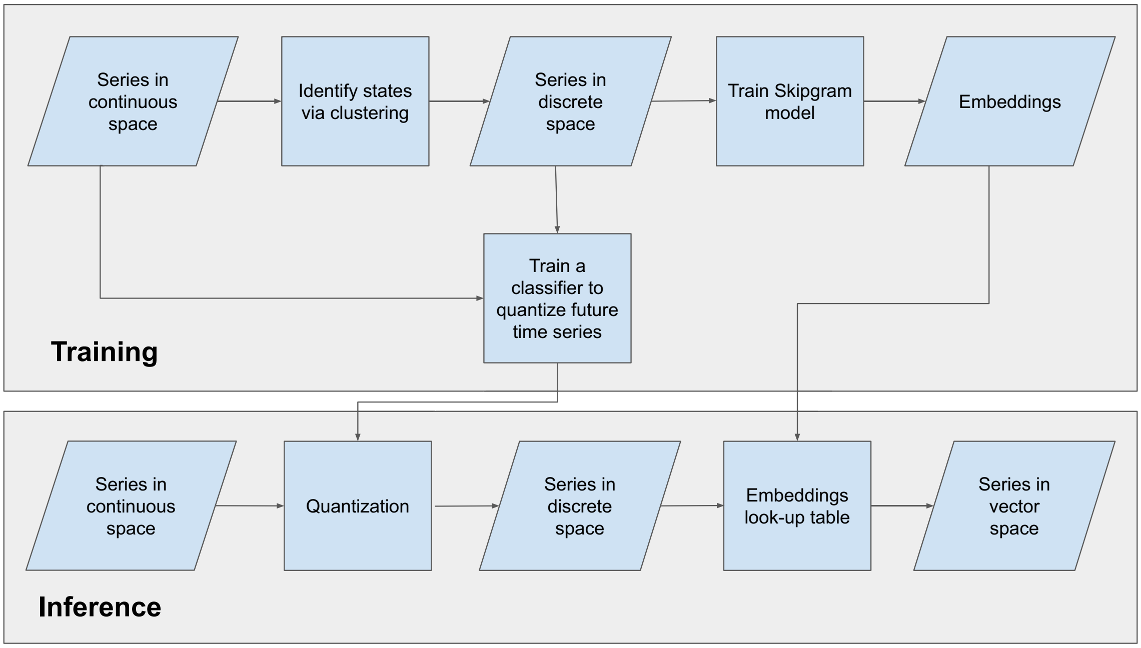

Fig. 4 Illustration of Signal2Vec algorithm

Definition 2 *Signal2Vec*: Let a time series be S of length N, a mapping function $f : R \rightarrow T^n$ and a lookup function $g : T^n \rightarrow V^{n*M}$, with T a finite set of n tokens and V an $M - dimensional$ vector space with n vectors. We define Signal2Vec as the mapping of a time series into a vector space with the following formula:

$$Signal2Vec = g \circ f : R \rightarrow V^{n*M} \tag{4}$$

Following the definition of Signal2Vec, the constructed vector space has as many vectors as the size of the set of tokens, with each vector having 300 dimensions. Thus, a time series is mapped to a sequence of tokens, which is mapped into a sequence of 300-dimensional vectors. Such a representation is not very helpful as it worsens the problem of dimensionality explosion. The concept of the average vector representation solves this problem.

Definition 3 *Average vector representation*: Let a time series S of length N mapped into a sequence of N vectors with $M - dimensions$ each. Let these vectors be $\hat{C}_i, i = 1, \ldots, N$. The average vector representation $S2V_{avg}$ of the time series S is defined by:

$$S2V_{avg} = 1/N \sum_{i=1}^{N} \hat{C}_i \tag{5}$$

Consequently, a time series of length N can be represented by just one $M - dimensional$ vector, as shown in Fig. 5. The benefit of this mapping is that the final representation is independent from the initial length N. In case a time series is very long, it can be broken into equal segments and the final vector representation will be the concatenation of the average vectors of each segment.

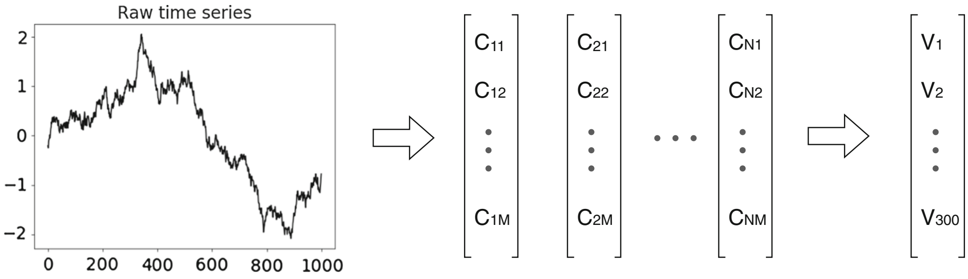

Fig. 5 Mapping a time series to one vector

Algorithm 1: Signal2Vec Training Trains a classifier for quantization and builds a vector space of the given time series.

```
Input: A time-series $ts\_data$ and the desired minimum and maximum
       states $min\_states$ and $max\_states$ respectively.
Output: A trained classifier which can be used to quantize relevant
        time-series and a dictionary to map the quantized values to
        vectors.
Data: Signal2vec training data.
$labels \leftarrow dict()$
$scores \leftarrow dict()$
$selected\_states \leftarrow 0$
$max\_score \leftarrow 0$
for $states \leftarrow min\_states$ to $max\_states$ do
    $clustering\_model \leftarrow KMeans(n\_clusters \leftarrow states)$
    $clustering\_model.fit(ts\_data)$
    $labels[states] \leftarrow clustering\_model.labels$
    $scores[states] \leftarrow silhouette\_score(ts\_data, labels)$
    if $scores[states] > max\_score$ then
        $max\_score \leftarrow scores[states]$
        $selected\_states \leftarrow states$
    end
end
$tokens \leftarrow labels[selected\_states]$
$classifier \leftarrow KNeighborsClassifier(n\_neighbors = selected\_states)$
$classifier.fit(ts\_data, tokens)$
$vocabulary \leftarrow set(tokens)$
$tokens\_sequence \leftarrow classifier.predict(ts\_data)$
$embeddings\_dict \leftarrow SkipGram.train(tokens\_sequence, vocabulary)$
return $classifier, embeddings\_dict$
```

The complete training and inference algorithms of Signal2Vec are described by Algorithms 1 and 2, respectively. Training Signal2Vec is fast because the neural net is very shallow and it does not slow inference because it happens only once. Therefore, inference time mainly depends on the inference time of the classifier as the mapping of tokens to embeddings is $O(1)$. Thus, the inference time complexity of k-NN is $O(N * S)$, where N is the length of the time series under transformation and S is the number of samples the classifier was trained. Considering that S does not change, we conclude that time complexity of Signal2Vec inference process is linear. Selecting a faster classifier is subject of future research and is advised for future implementations.

Algorithm 2: SIGNAL2VEC INFERENCE Transforms a given time series into a vector.

```
Input: A time-series $ts\_data$, a $classifier$ to quantize the given time
       series and a dictionary $embeddings\_dict$ to map the discrete
       sequence to a sequence of vectors.
Output: A vector representation $S2V_{avg}$ of the given time series.
Data: A time-series or a segment of any size.
$tokens\_sequence \leftarrow classifier.predict(ts\_data)$
$embeddings\_sequence \leftarrow list()$
foreach $token$ in $tokens\_sequence$ do
    $embeddings\_sequence.append(embeddings\_dict[token])$
end
$S2V_{avg} \leftarrow average(embeddings\_sequence)$
return $S2V_{avg}$
```

6 Experiments and results

The goal of the experiments is threefold: firstly, to show that dimensionality reduction of energy data leads to computationally light solutions for the task of multi-label NILM; secondly, to evaluate the effectiveness of eight different time series representations on this particular problem; and finally, to compare the best configuration of multi-NILM framework with the work of Tabatabaei et al. [51]. Due to the large number of parameters of a NILM system, e.g., sampling rate, different datasets, different time of the training and testing data and others, a direct comparison by using the results of other systems with different settings would not be fair. Therefore, the models proposed by Tabatabaei et al. are implemented from scratch, their experiments are replicated as close as possible and new ones are carried out for further investigation.

A synopsis of the three different types of experiments is shown in Table 1. It outlines the setup of the environment of each experiment and the goal that is pursued. More details are provided in the description of each experiment.

6.1 Data, metrics and methodology

The energy data that are used in the experiments come from two popular datasets UK-DALE [18] and REDD [23]. The latter one contains energy data from houses in the USA and the former one from UK houses. For parsing the data, we use NILMTK [8]. All experiments, including details of the implementation and results, are available at https://github.com/ChristoferNal/multi-nilm.

The examined time series, as mentioned in previous sections, are: PAA, SAX, 1d-SAX, DFT, SFA, BOSS, WEASEL and Signal2Vec. The only one that is suitable for transfer learning is Signal2Vec, and for this reason, we construct the embedding space only once. The vectors are produced from the aggregated energy signal of House 1 from UK-DALE dataset during the first 5 months of 2014 and with sampling rate 6s. As it is known from word2vec, the created embeddings generalize much better when trained with as much data as possible. However, due to limited energy data and to demonstrate the transfer learning property, the training set of the embeddings excludes the rest of the UK-DALE data and all the REDD dataset.

The machine learning models that are selected for the evaluation are limited to lightweight models supporting multi-label classification. These models include decision trees, extra trees, random forests and feedforward neural

Table 1 Summary of experiments

Experiment	Environment setup	Goal
Model selection	5 CV on House 1 from UK-DALE during first half of 2014	To select the best parameters and the best classifier for each representation
Evaluation	Train on House 1 from UK-DALE during 2013 and first half of 2014, test on House 1 of UK-DALE during 2nd half of 2014	To compare the different configurations of multi-NILM framework
Comparison with existing systems	Train/test on UK-DALE similar to evaluation. 5 CV on Houses 1 and 3 from REDD to replicate previous work	To compare the best multi-NILM configuration with previous solutions

networks. The parameters of the models were selected using grid search. Among these classifiers, the best ones were mostly variations of the neural network. In this series of experiments, we avoid using any advanced or complex algorithms like ensembles to keep the computational complexity as low as possible. This will make the models cost efficient in a cloud solution or efficient when running on a device with limited computation capabilities.

As far as the evaluation is concerned, the metrics that are used in multi-label classification tasks are micro- and macro-f1 scores. Macro-averaging firstly measures the performance of each class and then takes the mean, whereas micro-averaging calculates the overall performance [40]. The main difference is that macro-f1 emphasizes low performance of infrequent classes. Both measures are taken into consideration; however, macro-average is used as the main metric to compare different models.

6.2 Model selection

The methodology of model selection comprises pairing of each of the time series representations with all the candidate classifiers, taking into account the various parameters of both representation algorithms and classifiers. Overall, a few thousand different combinations are taken into consideration to avoid underrating any of the eight algorithms under evaluation. The model selection experiments are based on House 1 of UK-DALE dataset during the first half of 2014 and use a 5 cross-validation. The best machine learning models for each representation are presented in Tables 2 and 3. In these tables, the numbers in the parenthesis represent the number of trees for the tree-based models and number of neurons for the neural networks. A neural network can also have many layers, in which case the parenthesis includes the number of neurons per layer.

Power disaggregation depends a lot on the sampling rate or the length of the time frame that is used for inference. The sampling rate is kept stable at 6s. For the various time frames, different models and parameters are selected. The length of the time frames varies from 10 min to one day. Regarding Signal2Vec, the initial embeddings, which have been created at the sampling rate of 6 s, are not retrained. A retrain would be desirable though, as it is expected to capture different frequency features and increase the performance. The only parameter that is configurable for Signal2Vec is the number of average vectors per time frame. It has been found that longer time frames require more vectors. This is not a surprise as very long signals lead to summing many vectors which in turn results in obscured information. The rest of the algorithms are parameterized using grid search. For more details of the parameters, please refer to the related github repository. The appliances that are used during model selection are: oven, dish washer, fridge, washer dryer, boiler, vacuum cleaner, microwave, kettle, toaster, television, hair dryer and light.

6.3 Evaluation on UK-DALE

After model selection, the best machine learning models paired with the respective parameterized time series approximators are trained and tested. Training occurs in House 1 of UK-DALE dataset during March of 2013 to May of 2014. Testing takes place at the same house for the 12 appliances during the period June of 2014 to December of 2014. This is a very long period of testing, in contrast to the majority of the experiments in the literature, where NILM systems are tested for a period of a week or a couple of months. The results are presented in Fig. 6. The two most efficient time series representations are BOSS and Signal2Vec, with the second one demonstrating overall the best performance. The following two models are DFT and PAA, showing quite robust results, especially when the time window is long.

It is worth mentioning that Signal2Vec is the only representation which supports transfer learning. It was trained with a time frame equal to 6 s, and the same vectors are used for windows with duration from 10 min to one day. All other representations are adapted and parameterized for each different time frame. The only parameter of Signal2Vec is the number of average vectors per time frame which is 1 for short windows and 5 for the ones longer than 2 h. Another notable observation is that Signal2Vec not

Table 2 Model selection per time frame for Signal2Vec, BOSS, DFT and PAA

Period	Signal2Vec	BOSS	DFT	PAA
10 min	NN(1000)	NN(2000,100,100)	ExtraTrees(500)	ExtraTrees(500)
30 min	NN(1000,100)	NN(2000,100,100)	ExtraTrees(500)	ExtraTree
1 h	NN(1000,100)	NN(1000,100)	ExtraTrees(1000)	NN(2000,100,100)
2 h	NN(1000,100)	NN(2000,100)	ExtraTrees(1000)	NN(100,100,100)
4 h	NN(1000,100)	NN(1000,100)	ExtraTrees(200)	ExtraTrees(200)
8 h	NN(1000)	NN(2000,100)	ExtraTrees(2000)	ExtraTrees(200)
24 h	NN(1000,100)	NN(2000,100)	ExtraTrees(2000)	NN(1000,100))

Table 3 Model selection per time frame for SFA, 1d-SAX, SAX and WEASEL

Period	SFA	1d-SAX	SAX	WEASEL
10 min	NN(2000,100,100)	RandomForest(100)	NN(2000)	NN(1000)
30 min	NN(2000,100,100)	RandomForest(100)	NN(2000)	NN(1000)
1 h	NN(1000,2000,100)	ExtraTrees(100)	NN(100,)	NN(1000)
2 h	NN(2000,100,100)	ExtraTrees(100)	NN(100,)	NN(100)
4 h	ExtraTrees(500)	NN(1000,)	NN(1000,)	NN(100)
8 h	NN(100,50,100,50)	NN(100,100)	NN(2000)	NN(2000,100)
24 h	ExtraTrees(1000)	NN(1000,)	NN(100,100)	NN(1000,2000)

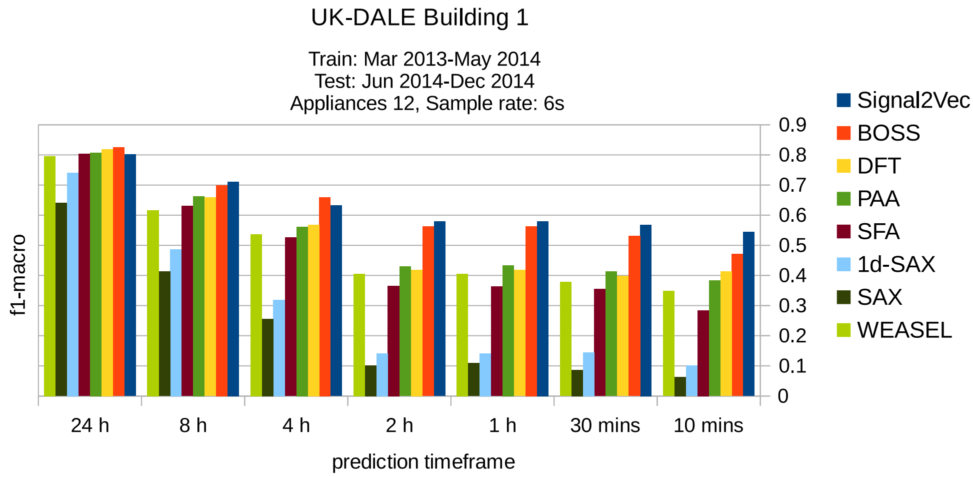

Fig. 6 Appliances: oven, microwave, dish washer, fridge, kettle, washer dryer, toaster, boiler, television, hair dryer, vacuum cleaner, light

only outperforms BOSS, but also uses a smaller neural network with almost half neurons.

Another important parameter in energy disaggregation is the number of devices that are recognized. This set of experiments includes 4 to 12 different appliances. Figure 7 presents some representative results with time period 1 h. The 12 appliances are the ones that are previously mentioned. The set of nine appliances contains microwave, dish washer, fridge, kettle, washer dryer, toaster, television, hair dryer and vacuum cleaner. The set of 6 is oven, dish washer, fridge, microwave, kettle and toaster. Finally, the set of four devices includes the ones that are mostly used in NILM research papers: dish washer, fridge, microwave and kettle. Signal2Vec again achieves the best f1 macro-score, with BOSS following. PAA, DFT and WEASEL are very close; SFA follows showing average performance, while SAX and 1d-SAX give the worse results.

Additionally, Table 4 presents both f1-macro and f1-micro, testing the disaggregation capabilities of two different sets of appliances of House 1 in UK-DALE dataset. One set includes high energy consumption devices such as oven, dish washer, fridge, washer dryer, boiler and vacuum cleaner. The other set includes low energy consumption devices such as microwave, kettle, toaster, television, hair dryer and light. As it is seen from the table, f1-macro and f1-micro most of the times are in good agreement. For the

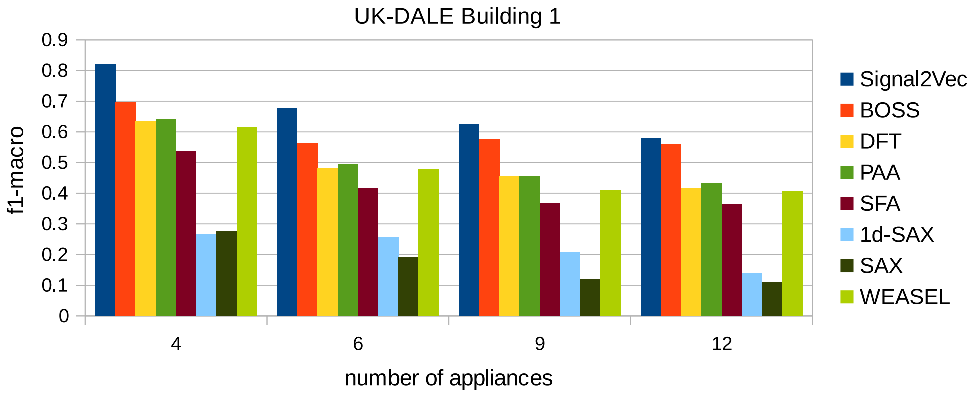

Fig. 7 Macro-f1 score per number of appliances, for experiments with 1-h time frame

Table 4 Results disaggregating high- and low-cost appliances

TS	High cost		Low cost	
Repr.	f1-macro	f1-micro	f1-macro	f1-micro
Signal2Vec	0.55 ± 0.004	0.77 ± 0.008	**0.61 ± 0.008**	**0.73 ± 0.012**
BOSS	**0.56 ± 0.003**	**0.79 ± 0.001**	0.57 ± 0.002	0.67 ± 0.004
PAA	0.42 ± 0.015	0.77 ± 0.010	0.44 ± 0.010	0.55 ± 0.009
DFT	0.41 ± 0.003	0.78 ± 0.001	0.44 ± 0.001	0.60 ± 0.001
WEASEL	0.42 ± 0.000	0.76 ± 0.003	0.41 ± 0.001	0.52 ± 0.004
SFA	0.35 ± 0.007	0.73 ± 0.005	0.38 ± 0.011	0.52 ± 0.014
1d-SAX	0.19 ± 0.000	0.74 ± 0.003	0.07 ± 0.004	0.22 ± 0.010
SAX	0.20 ± 0.031	0.69 ± 0.078	0.03 ± 0.053	0.12 ± 0.214

Values in bold correspond to the highest f score for each column

most costly appliances, this time BOSS gives slightly better results than Signal2Vec, whereas for the low-cost ones, Signal2Vec outperforms BOSS. A possible explanation of this result could be that high energy devices are easier to be recognized as they have a strong impact on the total power signal. On the other hand, low-consumption devices can easily be overlapped and thus it is difficult to trace them in the total signal. Given that BOSS is very robust to noise, we can assume that it sees the latter group of devices as noise. From the perspective of Signal2Vec, the results could be explained assuming that the embedding space has successfully learned the frequency that the two groups of devices are used, with the low-consumption ones being used in a daily basis.

To conclude, three different types of experiments have been presented. Multi-NILM framework proved to be more efficient when using Signal2Vec with a shallow neural network. It exceeded all other configurations of the framework not only in performance but also in computational efficiency. Only in a few cases, BOSS showed better performance. In order to enhance the significance of our results, a statistical test is also carried out. The hypothesis of the test is that the two versions of multi-NILM framework, based on Signal2Vec and BOSS accordingly, are equivalent, regardless the time window of the experiments. The hypothesis is tested with the Nadeau and Bengio correction to the paired Student t test [37], to avoid violating the key assumption of the Student's t test that the observations in each sample are independent. In order to reject or not the null hypothesis, *alpha* value is assumed to be 0.05. After calculating the p value equal to 0.0401, the null hypothesis can be rejected and the experimental results are statistically significant.

6.4 Comparison with existing multi-label NILM systems

According to the literature, the models proposed by Tabatabaei et al. are often used as a strong baseline to evaluate state-of-the-art solutions. Unfortunately, these experiments do not always take into account all the parameters of a NILM system setup and sometimes a comparison with the results of another research paper is not fair. In order to achieve an objective evaluation, the experiments that are described by Tabatabaei et al. are replicated as close as possible by developing the models from scratch. Additional experiments are also conducted to strengthen the results.

Tabatabaei et al. proposal has already been described in previous section and includes two very popular ensemble methods, named MLkNN and RAkEL. Both of them are evaluated extracting features either in time domain or in frequency domain. In time domain, the method that is used is time delay embeddings [16]. It has two parameters: the time delay and the dimensions of the embeddings. The first parameter is determined using the time-delayed mutual information method. The second one is configured via the method of false nearest neighbors. In frequency domain, the signal is transformed using the Haar wavelet.

Our setup is found to be very similar to the original one. Following the same methodology, the best time delay is determined to be 30 s and the best dimension equal to 6. These parameters apply for both REDD and UK-DALE. Apart from the techniques that are mentioned in the original paper, we also verified that these parameters give the best results by running fivefold cross-validation experiments for different values of the parameters. With respect to wavelet coefficients, they are selected so that they retain at least 95% of the signal energy. The sampling rate is 6s for all datasets and houses, and the window is set to 5 min. Houses 1 and 3 are selected from REDD. The appliances that are disaggregated in House 1 are oven, refrigerator, light, microwave, bath GFI, outlet and washer. For House 3, the appliances are electronics, furnace, washer dryer, microwave, bath GFI and kitchen outlet. House 1 from UK-DALE includes the largest variety of devices: oven, dish washer, fridge, washer dryer, boiler, vacuum cleaner, microwave, kettle, toaster, television, hair dryer and light.

Table 5 Comparing multi-NILM against Tabatabaei's et al. proposed models

Dataset	House	RAkEL Time	MLkNN Time	RAkEL Wavelet	MLkNN Wavelet	NN Signal2Vec	NN BOSS
REDD	1	0.476	0.490	0.450	0.504	**0.518**	0.369
REDD	3	0.372	0.414	0.312	0.434	**0.446**	0.170
UK-DALE	1	0.288	0.423	0.337	0.430	**0.492**	0.439

Values in bold correspond to the highest f score for each row

Table 6 Per appliance comparison on House 1 REDD

Appliance	RAkEL Time	MLkNN Time	RAkEL Wavelet	MLkNN Wavelet	NN Signal2Vec	NN BOSS
Oven	0.0	0.03	**0.12**	0.08	0.11	0.0
Refrigerator	0.27	**0.43**	0.19	**0.43**	**0.43**	0.41
Microwave	0.22	0.33	0.27	0.33	**0.45**	0.21
Washer dryer	0.09	0.29	0.11	0.28	**0.34**	0.04
Bath GFI	0.27	0.5	0.31	0.48	**0.55**	0.24
Sockets	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Light	0.39	0.43	0.4	0.41	**0.46**	0.22

Values in bold correspond to the highest f score for each row

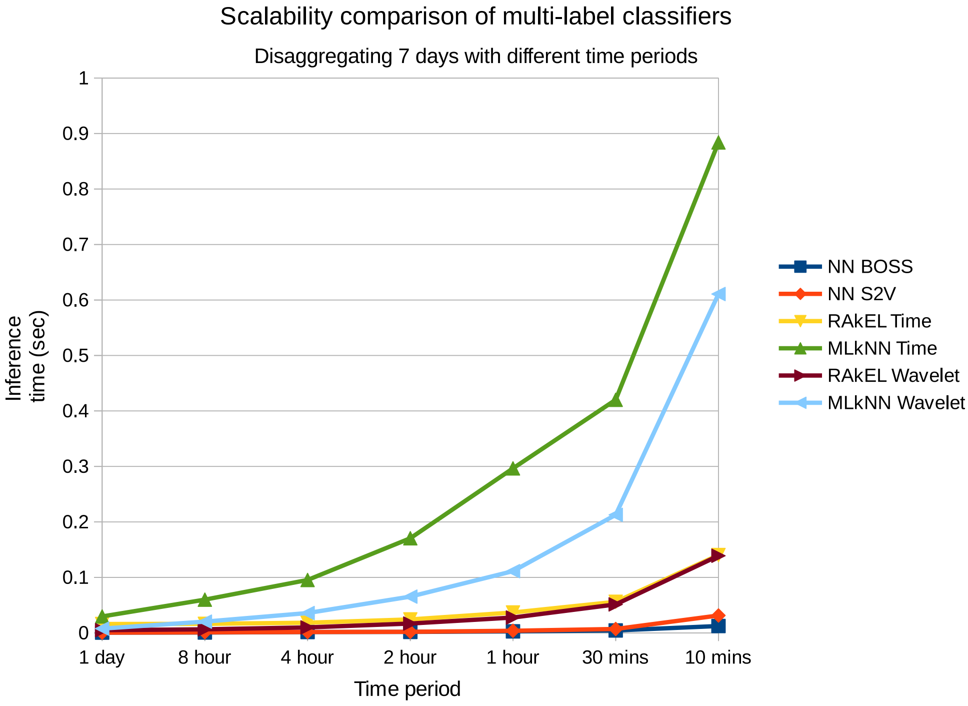

Fig. 8 Scalability comparison of multi-label NILM algorithms

The results of the experiments are summarized in Table 5 using macro-f1 score as a metric. The best multi-NILM model, based on Signal2Vec and a shallow neural network, outperformed all other models in all three houses. It is very important to emphasize that the vector space that is used by Signal2Vec, for the experiments on REDD dataset, is the same one trained on UK-DALE dataset. This is evidence of the generalization capabilities of the constructed embeddings. The properties that have been learnt from the energy consumption in a house in UK are also applicable to houses from the USA. The second best model is MLkNN using wavelets, but only for houses from REDD. On UK-DALE, the second best model is the one based on BOSS. On the other hand, BOSS does not show very consistent behavior as it shows the worst results for House 3 from REDD. In general, RAkEL shows lower f1 score when compared to MLkNN. A more detailed sample of the experimental results is presented in Table 6. The results come from REDD House 1 and represent the f1 score of each appliance. Signal2Vec achieves the best scores for all the appliances, apart from the oven. RAkEL in frequency domain performs the best score for the oven.

More confidence to our results is added using again the Nadeau and Bengio correction to the paired Student *t* test. The hypothesis now is that there is no statistically significant difference on the performance of MLkNN using wavelets and multi-NILM based on Signal2Vec, when the time window is 5 min. Given that *alpha* value is equal to 0.05, *p* value is calculated equal to 0.023. Therefore, the null hypothesis can be rejected and the experimental results are statistically significant.

Another aspect of this comparative analysis is to show how multi-NILM framework leads to computationally more efficient models. Figure 8 illustrates how each of the models scales in terms of the dimensionality of the data. The horizontal axis shows the window that has been used to train and test the models and ranges from 1 day to 10 mins. The vertical axis refers to the inference time of each classifier in seconds. The input to the models is energy data from REDD corresponding to a period of seven days. Therefore, smaller time window means more predictions within the seven days. Observing the diagram, MLkNN shows an exponential scalability on both time and frequency domains. RAkEL responds better with inference time being increased almost linearly for a period up to 30 min, after which it starts increasing exponentially. The two versions of multi-NILM framework perform the fastest inference, which is expected because they depend on very light classifiers and the input space is reduced by the respective time series approximator.

7 Conclusion

This paper revisits the problem of power disaggregation through the lens of multi-label classification task. A novel framework is proposed in order to tackle the problem of dimensionality explosion, reduce the cost of a NILM system and enable a cost-efficient solution. Eight popular time series representations are used for the first time in a NILM system. Signal2Vec demonstrates the most promising results and is the only model to support knowledge transfer. When compared to a strong baseline from the literature, Signal2Vec achieves the best macro-f1 score in all datasets and in a more efficient way.

Future research will focus on improving the model of Signal2Vec and increase the performance of multi-NILM framework. A key idea to improve Signal2Vec is to improve its quantization process by using an alternative method of clustering, such as Gaussian mixture models. The entire quantization process could also be replaced by an algorithm which transforms a time series into words like PAA, and then, apply the skip-gram model on top of this transformation. Regarding Multi-NILM framework, new experiments are advised to be designed using other innovative dimensionality reduction methods. More advanced multi-label classifiers, like deep neural networks, should also be considered, but always with the goal to find a balance between efficiency and accuracy.

The exponential increase in connected devices makes it mandatory to look for low-cost solutions that can run on low end devices. Researchers should focus on systems which, apart from achieving better results, should also be practically viable.

Acknowledgements This research has been co-financed by the European Regional Development Fund of the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH–CREATE–INNOVATE (Project Code: 95699—Energy Controlling Voice Enabled Intelligent Smart Home Ecosystem). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Agrawal R, Faloutsos C, Swami A (1993) Efficient similarity search in sequence databases. In: International conference on foundations of data organization and algorithms. Springer, pp 69–84
2. Armel KC, Gupta A, Shrimali G, Albert A (2013) Is disaggregation the holy grail of energy efficiency? The case of electricity. Energy Policy 52:213–234
3. Baranski M, Voss J (2003) Nonintrusive appliance load monitoring based on an optical sensor. In: 2003 IEEE Bologna power tech conference proceedings, vol 4. IEEE, pp. 8
4. Basu K, Debusschere V, Bacha S (2012) Load identification from power recordings at meter panel in residential households. In: 2012 XXth international conference on electrical machines. IEEE, pp 2098–2104
5. Basu K, Debusschere V, Bacha S (2013) Residential appliance identification and future usage prediction from smart meter. In: IECON 2013-39th annual conference of the IEEE industrial electronics society. IEEE, pp 4994–4999
6. Basu K, Debusschere V, Bacha S, Maulik U, Bondyopadhyay S (2014) Nonintrusive load monitoring: a temporal multilabel classification approach. IEEE Trans Ind Inform 11(1):262–270
7. Basu K, Debusschere V, Douzal-Chouakria A, Bacha S (2015) Time series distance-based methods for non-intrusive load monitoring in residential buildings. Energy Build 96:109–117

8. Batra N, Kelly J, Parson O, Dutta H, Knottenbelt W, Rogers A, Singh A, Srivastava M (2014) Nilmtk: an open source toolkit for non-intrusive load monitoring. In: Proceedings of the 5th international conference on Future energy systems. ACM, pp 265–276
9. Batra N, Kukunuri R, Pandey A, Malakar R, Kumar R, Krystalakos O, Zhong M, Meira P, Parson O (2019) Towards reproducible state-of-the-art energy disaggregation. In: Proceedings of the 6th ACM international conference on embedded systems for energy-efficient built environments (BuildSys' 19). ACM, New York, NY, USA
10. Chang HH, Chien PC, Lin LS, Chen N (2011) Feature extraction of non-intrusive load-monitoring system using genetic algorithm in smart meters. In: 2011 IEEE 8th international conference on e-business engineering. IEEE, pp 299–304
11. Duchi J, Hazan E, Singer Y (2011) Adaptive subgradient methods for online learning and stochastic optimization. J Mach Learn Res 12:2121–2159
12. Egarter D, Pöchacker M, Elmenreich W (2015) Complexity of power draws for load disaggregation. arXiv preprint arXiv:1501.02954
13. Froehlich J, Larson E, Gupta S, Cohn G, Reynolds M, Patel S (2010) Disaggregated end-use energy sensing for the smart grid. IEEE Pervasive Comput 10(1):28–39
14. Gutmann MU, Hyvärinen A (2012) Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. J Mach Learn Res 13:307–361
15. Hart GW (1992) Nonintrusive appliance load monitoring. Proc IEEE 80(12):1870–1891. https://doi.org/10.1109/5.192069
16. Kantz H, Schreiber T (2004) Nonlinear time series analysis, vol 7. Cambridge University Press, Cambridge
17. Kato T, Cho HS, Lee D, Toyomura T, Yamazaki T (2009) Appliance recognition from electric current signals for information-energy integrated network in home environments. In: International conference on smart homes and health telematics. Springer, pp 150–157
18. Kelly J, Knottenbelt W (2015) The UK-dale dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes. Sci Data 2:150007
19. Keogh E, Chakrabarti K, Pazzani M, Mehrotra S (2001) Dimensionality reduction for fast similarity search in large time series databases. Knowl Inf Syst 3(3):263–286
20. Kim JG, Lee B (2019) Appliance classification by power signal analysis based on multi-feature combination multi-layer lstm. Energies 12(14):2804
21. Klemenjak C, Makonin S, Elmenreich W (2020) Towards comparability in non-intrusive load monitoring: on data and performance evaluation. In: 2020 IEEE power & energy society innovative smart grid technologies conference (ISGT)
22. Klemenjak C, Reinhardt A, Pereira L, Makonin S, Bergés M, Elmenreich W (2019) Electricity consumption data sets: Pitfalls and opportunities. In: Proceedings of the 6th ACM international conference on systems for energy-efficient buildings, cities, and transportation, pp 159–162
23. Kolter JZ, Johnson MJ (2011) Redd: a public data set for energy disaggregation research. In: Workshop on data mining applications in sustainability (SIGKDD), San Diego, CA, vol 25, pp 59–62
24. Krystalakos O, Nalmpantis C, Vrakas D (2018) Sliding window approach for online energy disaggregation using artificial neural networks. In: Proceedings of the 10th Hellenic conference on artificial intelligence. ACM, p 7
25. Lai YX, Lai CF, Huang YM, Chao HC (2013) Multi-appliance recognition system with hybrid SVM/GMM classifier in ubiquitous smart home. Inf Sci 230:39–55
26. Laughman C, Lee K, Cox R, Shaw S, Leeb S, Norford L, Armstrong P (2003) Power signature analysis. IEEE Power Energy Mag 1(2):56–63
27. Li D, Dick S (2016) Whole-house non-intrusive appliance load monitoring via multi-label classification. In: 2016 international joint conference on neural networks (IJCNN). IEEE, pp 2749–2755
28. Li D, Dick S (2019) Residential household non-intrusive load monitoring via graph-based multi-label semi-supervised learning. IEEE Trans Smart Grid 10(4):4615–4627
29. Li D, Sawyer K, Dick S (2015) Disaggregating household loads via semi-supervised multi-label classification. In: 2015 annual conference of the North American fuzzy information processing society (NAFIPS) held jointly with 2015 5th world conference on soft computing (WConSC). IEEE, pp 1–5
30. Liang J, Ng SK, Kendall G, Cheng JW (2009) Load signature study—part i: Basic concept, structure, and methodology. IEEE Trans Power Deliv 25(2):551–560
31. Lin J, Keogh E, Wei L, Lonardi S (2007) Experiencing sax: a novel symbolic representation of time series. Data Min Knowl Discov 15(2):107–144
32. Lin J, Khade R, Li Y (2012) Rotation-invariant similarity in time series using bag-of-patterns representation. J Intell Inf Syst 39(2):287–315
33. Loukas EP, Bodurri K, Evangelopoulos P, Bouhouras AS, Poulakis N, Christoforidis GC, Panapakidis I, Chatzisavvas KC (2019) A machine learning approach for nilm based on odd harmonic current vectors. In: 2019 8th international conference on modern power systems (MPS). IEEE, pp 1–6
34. Malinowski S, Guyet T, Quiniou R, Tavenard R (2013) 1d-sax: a novel symbolic representation for time series. In: International symposium on intelligent data analysis. Springer, pp 273–284
35. Marchiori A, Hakkarinen D, Han Q, Earle L (2010) Circuit-level load monitoring for household energy management. IEEE Pervasive Comput 10(1):40–48
36. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp 3111–3119
37. Nadeau C, Bengio Y (2000) Inference for the generalization error. In: Advances in neural information processing systems, pp 307–313
38. Nalmpantis C, Vrakas D (2019) Machine learning approaches for non-intrusive load monitoring: from qualitative to quantitative comparation. Artif Intell Rev 52(1):217–243
39. Nalmpantis C, Vrakas D (2019) Signal2vec: time series embedding representation. In: International conference on engineering applications of neural networks. Springer, pp 80–90
40. Rak R, Kurgan L, Reformat M (2005) Multi-label associative classification of medical documents from medline. In: Fourth international conference on machine learning and applications (ICMLA'05). IEEE, pp 8
41. Read J, Pfahringer B, Holmes G, Frank E (2011) Classifier chains for multi-label classification. Mach Learn 85(3):333
42. Restuccia F, D'Oro S, Melodia T (2018) Securing the internet of things in the age of machine learning and software-defined networking. IEEE Internet Things J 5(6):4829–4842
43. Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 20:53–65
44. Ruzzelli AG, Nicolas C, Schoofs A, O'Hare GM (2010) Realtime recognition and profiling of appliances through a single electricity sensor. In: 2010 7th annual IEEE communications society conference on sensor, mesh and ad hoc communications and networks (SECON). IEEE, pp 1–9

45. Schäfer P (2015) The boss is concerned with time series classification in the presence of noise. Data Min Knowl Discov 29(6):1505–1530
46. Schäfer P, Högqvist M (2012) Sfa: a symbolic Fourier approximation and index for similarity search in high dimensional datasets. In: Proceedings of the 15th international conference on extending database technology. ACM, pp 516–527
47. Schäfer P, Leser U (2017) Fast and accurate time series classification with weasel. In: Proceedings of the 2017 ACM on conference on information and knowledge management. ACM, pp 637–646
48. Singhal V, Maggu J, Majumdar A (2018) Simultaneous detection of multiple appliances from smart-meter measurements via multi-label consistent deep dictionary learning and deep transform learning. IEEE Trans Smart Grid 10(3):2969–2978
49. Srinivasan D, Ng W, Liew A (2005) Neural-network-based signature recognition for harmonic source identification. IEEE Trans Power Deliv 21(1):398–405
50. Suzuki K, Inagaki S, Suzuki T, Nakamura H, Ito K (2008) Nonintrusive appliance load monitoring based on integer programming. In: 2008 SICE annual conference. IEEE, pp 2742–2747
51. Tabatabaei SM, Dick S, Xu W (2016) Toward non-intrusive load monitoring via multi-label classification. IEEE Trans Smart Grid 8(1):26–40
52. Takens F (1981) Detecting strange attractors in turbulence. In: Dynamical systems and turbulence, Warwick 1980. Springer, pp 366–381
53. Tsoumakas G, Katakis I (2007) Multi-label classification: an overview. Int J Data Wareh Min 3(3):1–13
54. Verma S, Singh S, Majumdar A (2019) Multi label restricted Boltzmann machine for non-intrusive load monitoring. In: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 8345–8349
55. Wu X, Kumar V (2009) The top ten algorithms in data mining. CRC Press, Boca Raton
56. Zhang C, Zhong M, Wang Z, Goddard N, Sutton C (2018) Sequence-to-point learning with neural networks for non-intrusive load monitoring. In: Thirty-second AAAI conference on artificial intelligence
57. Zhang ML, Zhou ZH (2007) Ml-knn: a lazy learning approach to multi-label learning. Pattern Recognit 40(7):2038–2048
58. Zhong M, Goddard N, Sutton C (2015) Latent Bayesian melding for integrating individual and population models. In: Advances in neural information processing systems, pp 3618–3626
59. Zia T, Bruckner D, Zaidi A (2011) A hidden Markov model based procedure for identifying household electric loads. In: IECON 2011-37th annual conference of the IEEE industrial electronics society. IEEE, pp 3218–3223

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.