ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230638

A Neuro-Symbolic Approach for Non-Intrusive Load Monitoring

Gianluca Apriceno[ab*], Luca Erculiani[b*] and Andrea Passerini[b]

[a]Fondazione Bruno Kessler
[b]University of Trento

Abstract. A requirement of Smart Grids is the ability to predict the energy consumption patterns of their users. In the residential domain, this is usually not feasible due to the inability of the grid to dialog with (legacy) domestic appliances. To overcome this issue Non Intrusive Load Monitoring (NILM) was introduced, a task in which a predictor is used to disaggregate household power consumption. Many of the newer approaches make use of Neural Networks to accomplish this task, due to their superior ability to detect patterns in temporal (thus sequential) data. These models unfortunately require a huge amount of data to achieve good performance, and have the tendency to overfit the training data, making them difficult to predict future consumptions. For these reasons, adapting them to optimally predict a (future) house's consumption requires expensive and often prohibitive data collection phases. We propose a solution in the form of a neuro-symbolic framework that refines neural network predictions via a constrained optimization problem modelling the characteristics of the appliances of a house. This combined approach achieves superior performance with respect to the neural network alone over two out of five appliances and comparable results for the remaining ones, without requiring further training data.

1 Introduction

The past few years have seen an increased awareness by people and institutions on climate change and its consequences. With the objective to avoid/limit its effects, people have started to change their habits (e.g. walk and cycle more or drive electric vehicles), while governments have committed themselves to reduce CO2 emissions [1] with the long term goal to be climate-neutral by 2050[2]. Therefore, huge investments have been made in renewable (aka green) energies in order to meet the increasing demand and gradually replace the dependence on traditional source of energies [3]. Nevertheless, renewable energies have a limitation connected to the presence of their natural source. Therefore, investments in the development of a "smart" electrical network, which is able to guarantee an effective distribution of energy, have been made in parallel. This network has been called smart grid and, differently to the traditional distribution, where there is an unidirectional flow from producer to consumer, a bidirectional exchange of information is achieved in order to guarantee an effective usage of the electricity.

One of the requirements of a smart grid consists in understanding and predicting the energy consumption pattern of the consumers. This is done by processing the data of the energy consumption of the individual appliances of all houses. To measure the energy consumption of the appliances, two approaches can be adopted: Intrusive Load Monitoring (ILM) and Non-Intrusive Load Monitoring (NILM). ILM is based on the installation of a sensor (e.g., smart plugs or smart sockets) for each appliance that monitors and sends information about the consumption of the appliance back. Even though, ILM ensures accurate measurements, it is usually expensive and often perceived as "too intrusive" by consumers (i.e., each sensor has to be installed inside the house of the consumer). On the contrary, NILM, where the consumption of each appliance is obtained from the disaggregation of the total energy of the house, represents a cheap and less invasive solution (even if less accurate) with respect to ILM.

Due to the success that deep learning models have achieved in solving tasks of different domains (e.g. computer vision and natural language processing), these models have been started to be applied in NILM. Nevertheless, these models require a huge amount of annotated training data and lack of the ability to generalize to unseen situations (e.g., situations not included in the training data). Therefore, in the last years, neuro-symbolic techniques, which combine neural networks with symbolic reasoning, have started to be applied to overcome these issues.

In this paper, we devise a neuro-symbolic algorithm that combines the prediction of the Neural Network with a Constrained Programming optimization problem, used to refine the raw prediction of the network. The optimization problem is used to encode the behavior of the appliance in terms of its consumption over time. For example, the characteristics that can be captured may be:

- Minimum/maximum duration of appliance activation.
- Minimum/maximum instantaneous used power.
- Presence of multiple "states" with different duration and power adsorption.

All the characteristics, which are encoded as logic formulas, can then be used to correct the output of the network, and to find coherent intervals of times in which the the appliance was in use (the whole duration of a washing cycle of a washing machine, for instance), together with their expected consumption. Furthermore, the addition of this logic layer allows for the disambiguation of dubious predictions done by the Neural Network (due for instance to the presence of noise), increasing the overall disaggregation performance. Experiments have been performed on the UK-DALE dataset in a "seen" setting, and show that our combined approach is able to outperform a fully neural model over the prediction of two out of five appliances, with comparable results for the remaining ones.

The rest of the paper is organized as follows: Section 2 briefly re-

* These two authors contributed equally
[1] Paris agreement
[2] Net-zero
[3] Global electricity insights 2022

views the state of the art on NILM, focusing mostly on deep learning approaches; Section 3 formally describes the problem; Sections 4 describes our proposed approach; Section 5 presents the experimental setting; Section 6 shows the experimental results; Finally, in section 7 conclusions are drawn and directions for future works are briefly discussed.

2 State of the art

Since its introduction by Hart [12], NILM has gathered the attention of researchers not only for the intrinsic challenges [13] that it involves but for the benefits that approach like NILM could bring into our everyday life [10]. Over the last decade, more "classical" approaches to NILM (see [24] for a survey) have been replaced by deep learning methods. This is due to the success that deep learning has achieved in many different fields like computer vision and natural language processing (see [6] and [19] for a survey). In [25], authors instantiate a network (one for each appliance) and train it to learn a mapping between a sequence of mains to a sequence of appliance consumption. At inference time, multiple predictions for a generic time t are averaged in order to obtain a single prediction. [15] obtains more accurate results with respect to [25], by predicting from the current window of mains only the consumption of the appliance which corresponds to the middle point of the window. A more recent approach [20], adopts an attention mechanism to improve the generalization capability of the overall model. [18] proposes a novel architecture which integrates the Fourier transform and achieves comparable results with respect to the state art approaches while being faster and smaller. In all the aforementioned approaches, prior (explicit) knowledge about the behaviour of the appliances is not exploited to perform the disaggregation (i.e. the disaggregation is completely learnt from data). As far as we know, the only attempt to exploit prior knowledge to solve NILM has been done by [5], but this is a fully symbolic approach and then no neural networks have been used. In the last years, neuro-symbolic integration, which integrates neural and symbolic AI, has emerged as a new paradigm to merge the strengths and reduce/limit the weaknesses of the neural and symbolic "worlds" [14]. Therefore, different neuro-symbolic frameworks have been proposed over the years like frameworks based on fuzzy-logic [3, 17], probabilistic logic programming [16, 23] or Optimization modulo theories [21]/Mixed Integer Linear Programming (MILP) encoding (see [9] and [2]). Furthermore, these approaches have been applied to solve complex tasks like semantic image interpretation [8] and event recognition from different data sources (e.g. video [1, 2] and audio [22]). Inspired by [2], we encode the background knowledge about the behaviour of each appliance as a mixed integer linear programming problem (MILP), and use it to refine the prediction of the neural network. Differently from [2], we also learn the parameters related to the MILP problem (see section 4 for details).

3 Problem definition

NILM consists in the disaggregation of the total energy consumption of a house into the consumption of the individual appliances belonging to it. Formally, denoting with $X_{tot} = (x_1, \dots x_t)$, $x_i \in \mathbb{R}^+$, the total energy consumption for the period of time starting at 1 and ending at t, and supposing the presence of n appliances, we can express the total energy consumption at a given time i as:

$$x_i = \sum_{j=1}^{n} y_{ji} + \gamma_i \quad with\ 1 \leq i \leq t$$

where y_{ji} is the consumption of the $j-th$ appliance at time i and γ_i is a noise factor. We are interested in finding all the appliance consumption $Y_j = (y_1, \dots, y_t)\ y_i \in \mathbb{R}^+$, from X_{tot}. To achieve this objective, we also assume to have a background knowledge $\mathcal{K}$ (defined over a first order language $\mathcal{L}$) about the consumption/behaviour of each appliance. Therefore, our problem consists in finding an interpretation $\mathcal{I}$ (i.e. predicting a sequence of values, one for each appliance), such that $\mathcal{I} \models \mathcal{K}$.

Example 1 *Suppose that for a given house h, we have that* $X^h_{tot} = \{1000, 1000, 1200, 1400\}$. *We want to predict the consumption for the appliance* app_1 *of h. The background knowledge* $\mathcal{K}$ *states that when* app_1 *is active, its consumption has to be less than* 500 *at time t and the sum of consumption of the next two timestamps (i.e,* $t+1$ *and* $t+2$*) has to be between* 700 *and* 1400*. We can write a first order logical formula that expresses the above conditions:*

$$\begin{aligned}
\forall t_s, t_e \exists t_i, t_j, t_z \\
& cons(x, t_i, y_{t_i}) \leq 500 \wedge \\
& 700 \leq cons(x, t_j, y_{t_j}) + cons(x, t_z, y_{t_z}) \leq 1400 \wedge \\
& active(x, t_i)\ \wedge\ active(x, t_j) \wedge\ active(x, t_z) \wedge \\
& t_s \leq t_i < t_j < t_z \leq t_e \wedge \\
& t_j = t_i + 1 \wedge t_z = t_i + 2
\end{aligned}$$

where t_s *and* t_e *represent respectively the begin and the end of the period of consumption,* $cons(x, t, y_t)$ *is a function that returns the consumption of a generic appliance* x *at time* t *(i.e., it returns* y_t*) and active is a predicate that returns 1 if an appliance is active at a time t and 0 otherwise. Continuing the example, if we know that* app_1 *is active between time 2 and 4, some of the interpretations that satisfy* $\mathcal{K}$ *are:*

$$\begin{aligned}
\mathcal{I}_1 = \{&cons(app_1, 2, 250), cons(app_1, 3, 500), cons(app_1, 4, 600), \\
&active(app_1, 2), active(app_1, 3), active(app_1, 4)\} \\
\mathcal{I}_2 = \{&cons(app_1, 2, 350), cons(app_1, 3, 600), cons(app_1, 4, 700), \\
&active(app_1, 2), active(app_1, 3), active(app_1, 4)\} \\
&\vdots
\end{aligned}$$

As can be seen, there may be more than one interpretation that satisfies $\mathcal{K}$ *(we denote with* $\mathcal{I}_c$ *the set of such interpretations). Therefore, as done in [2], we introduce a cost function c that gives a score (i.e. a real value) for each* $\mathcal{I}$ *and select the interpretation with the minimum cost:*

$$\mathcal{I}_{\min} = \operatorname*{argmin}_{\mathcal{I} \in \mathcal{I}_c} c(\mathcal{I})$$

To find $\mathcal{I}_c$, we devise a neuro-symbolic approach where the initial prediction of the network Y_{NN_j} is refined in order to produce a new prediction $Y^*_{NN_j}$ that keeps into account the knowledge $\mathcal{K}$. To train the overall system, we have a training set of consumption of m houses:

$$D = \{(X^i_{tot}, \{Y^i_j\}^{n_i}_{j=1})\}^m_{i=1}$$

where Y^i_j denotes the consumption of the $j-th$ appliance in house i and n_i the number of appliances in the same house (i.e. different houses may have different appliances). It is not always that the entire D is used for the training phase (further details follow).

4 Proposed approach

The proposed approach consists in a neuro-symbolic framework that combines the generalization capabilities of a Neural Network with a constrained satisfaction problem (CSP), which has built in its specifications the domain knowledge and enables to refine the predictions of the network (see figure 1 for a high level overview of the proposed approach). Differently from the approaches found in literature [4,11] the neural and symbolic models are not in competition but cooperate together. The framework gets an initial prediction from the neural network, which is then refined by the optimization problem. Contrary to some other approaches, our CSP problem is only partially defined, and before being used is meant to be trained onto a small set of labeled examples of activations of a specific appliance to fit it properly.

The goal of the CSP is to model the energy consumption patterns of a specific appliance. Before moving on, we must briefly formalize the expected pattern of the appliance. We can imagine the lifecycle as a contiguous infinite sequence of idle intervals (in which the appliance is not used/switch off/in standby) and activation intervals, in which most of the power gets consumed and some useful work is performed. Each activation j starts at a certain time t^j_{start} and ends at time t^j_{end} (we will use t_{start} and t_{end} later in the article when referring to a generic activation). An example of this behavior is depicted in Figure 2.

Following previous work on the field [7], we model an appliance as a finite state automata. Indeed, an appliance is a machine that is built to perform a predetermined sequence of actions cyclically. Therefore, a neuro-symbolic approach, which models an appliance's consumption as a sequence of states, has the potential to improve the accuracy of the prediction. In addition, a neural network trained to predict an appliance's consumption over time, generally reasons in terms of time-points (i.e., the consumption at time x is y), while our proposed neuro-symbolic approach reasons in term of intervals of time (i.e., for x seconds the consumption will follow a specific trend curve) making the prediction more coherent over time. In this conceptualization, the appliance is represented as a collection of n states $\{s_i\}^n_1$ each associated with a duration t_i and a function $f_i : \mathbb{R}^+ \rightarrow \mathbb{R}$ that maps each instant of the interval t_i with the power consumed by the appliance at that instant. Thus, the activation cycle of the appliance is described by a set of states $S = \{< s_i, t_i, f_i >\}^n_{i=1}$. There is then a special state s_{idle}, the *initial state*, with associated its function f_{idle}, that has no fixed duration. This is the state where the appliance is before is switched on and after is switched off (or put in stand-by).

A life cycle of an appliance starts in $s_0 = s_{idle}$, when it is activated it switches to s_1 and stays in that state for t_1 time. After that, the appliance moves to s_2 for t_2 time and then switches to the next state. After t_N time spent in state s_N, the appliance shuts off and goes back to s_{idle}.

Our CSP framework applies these ideas by first learning a set of states S, by fitting a small number of examples of consumption patterns of the target. These are then used to refine the predictions of the network for the time in which the appliance is considered active *i.e.* between t_{start} and t_{end}. We currently do not model the idle state s_0 (the CSP problem "knows" that the power consumption at idle is some constant value p_{idle}), relying on the neural network to (roughly) identify the t_{start} and t_{end} of each activation. In the next sections, we will describe in detail both the training and inference procedure.

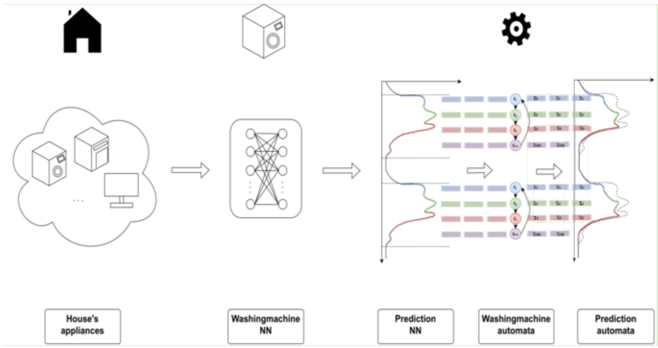

Figure 1: Overall approach on washingmachine appliance: the total consumption of the house (i.e., the total consumption of all its appliances) is passed to the washingmachine NN that provides an initial prediction for the consumption of the washingmachine. This prediction is then refined by its corresponding (learnt by CSP) automata that changes the prediction of the washingmachine NN by leveraging the knowledge encoded in each of its state.

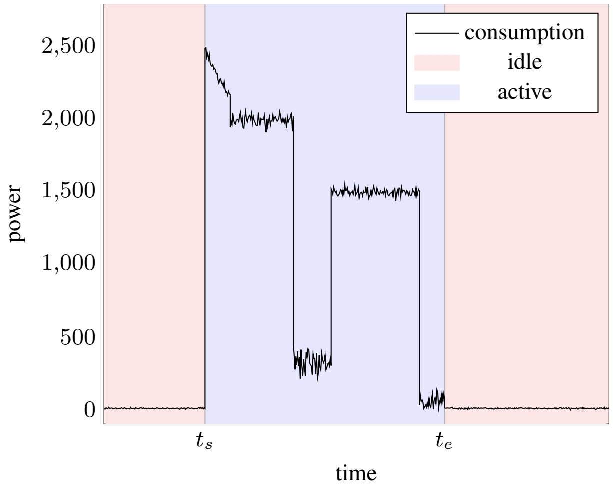

Figure 2: Example of an activation cycle of an appliance.

4.1 Training

As explained in previous sections, the training procedure of the CSP problem relies on labeled data about a few activations of the appliance. We can formalize it as a function $P : \mathbb{R}^+ \rightarrow \mathbb{R}$ that links each time instant with a power consumption. For training, we have a series of m intervals $\{< t_{j,start}, t_{j,end} >\}^m_{j=1}$ that encode the boundaries of the activations. The last bit of information is the constant idle power consumption p_{idle} of the appliance. This is necessary because, albeit the problem relies on the network for the prediction of power consumption when idle, it cannot assume that the activation intervals are perfectly timed, thus it must account for the appliance *potentially* being in idle near t_{start} and t_{end}. Figure 3 depicts the whole activation cycle of an appliance that follows these principles. The target is modeled using 3 active states s_1, s_2, s_3, with their corresponding state functions f_1, f_2, f_3. Aside from the "active" states, the framework uses two more states s_0 and s_{n+1} to model the idle state (before the first and after the last active state), both of them represented modeled by the function f_{idle}, that have a duration of t_0 and t_{n+1}. The objective of the fitting is to obtain a curve (by adding up state and idle functions) that closely matches the target consumption curve in the interval between t_{start} and t_{end}.

In order to formalize the fitting problem, we must introduce some constructs. We define $t^*_{j,i}$ as the summation of $t_{j,start}$ and of all the durations of all the $i-1$ states:

$$t^*_{j,i} = t_{j,start} + \sum_{k=1}^{i-1} t_{j,k}$$

Each function f_i is an exponential function in the form:

$$f_{j,i}(t) = \alpha_{j,i}e^t + \beta_{j,i}$$

while f_{idle} has the form:

$$f_{j,idle}(t) = C$$

The function that is learnt from a specific activation j is then:

$$f_j(t) = \begin{cases} f_{j,idle}(t) & t_{j,start} < t \leq t^*_{j,1} \\ f_{j,1}(t) & t^*_{j,1} < t \leq t^*_{j,2} \\ \ldots \\ f_{j,i}(t) & t^*_{j,i} < t \leq t^*_{j,i+1} \\ \ldots \\ f_{j,n}(t) & t^*_{j,n-1} < t \leq t^*_{j,n} \\ f_{j,idle}(t) & t^*_{j,n} < t \leq t_{j,end} \end{cases} \quad (1)$$

Defining the function encoding the interval of interest as

$$P_j(t) = \begin{cases} P(t - t_{j,start}) & t_{j,start} < t < t_{j,end} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

we can define the difference between the real and predicted consumption as

$$\Delta c_j = \int_{t_{j,start}}^{t_{j,end}} |P_j(t) - f_j(t)| dt \quad (3)$$

Moreover, the training problem tries to minimize the deviation between the parameters. In particular the deviation between the duration of the states across the various training sequences:

$$\Delta s = \sum_{i=1}^{n} \max(\{t_{j,i}\}_{j=1}^{m}) - \min(\{t_{j,i}\}_{j=1}^{m}) \quad (4)$$

and the same deviation for the $f_{j,i}(t)$ parameters $\alpha_{j,i}$ and $\beta_{j,i}$:

$$\Delta \alpha = \sum_{i=1}^{n} \max(\{\alpha_{j,i}\}_{j=1}^{m}) - \min(\{\alpha_{j,i}\}_{j=1}^{m}) \quad (5)$$

$$\Delta \beta = \sum_{i=1}^{n} \max(\{\beta_{j,i}\}_{j=1}^{m}) - \min(\{\beta_{j,i}\}_{j=1}^{m}) \quad (6)$$

The overall training problem is then defined as:

$$\underset{C,\alpha_{j,i},\beta_{j,i},t_{j,i}}{\text{minimize}} \quad \Delta s + \Delta \alpha + \Delta \beta + \sum_j \Delta c_j \quad (7)$$

4.2 *Inference*

Once the parameters $C, \alpha_{j,i}, \beta_{j,i}, t_{j,i}$ have been optimized in training, a new optimization problem is defined for inference. Differently from training, this time there are two distinct sources of input data: the actual raw aggregated power consumption (the same input given

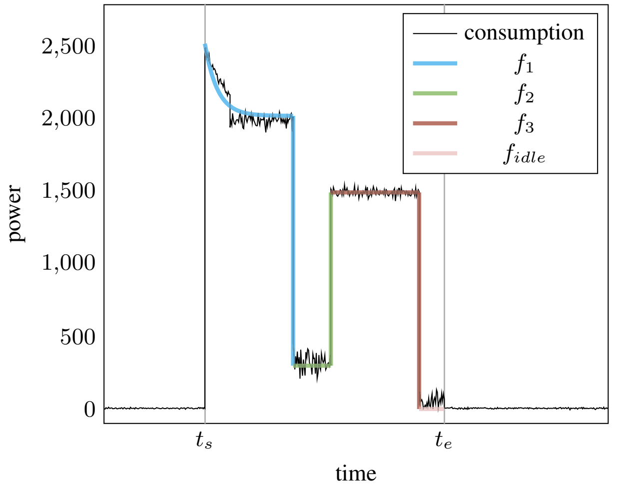

Figure 3: Representation of the result of the training procedure. The target activation was fitted using an automata with 3 states s_1, s_2, s_3. In the figure are depicted the corresponding state functions f_1 (in cyan), f_2 (green), f_3 (red) and the idle function (a constant value, in this case 0) f_{idle} for the idle state.

to the neural network) and the output of the neural network. The problem is expected to refine the neural output by using the "learnt behaviour" of the appliance (encoded in $C, \alpha_{j,i}, \beta_{j,i}, t_{j,i}$), while at the same time ensuring that the final prediction does not conflict with the actual instantaneous aggregated power consumption (e.g., predicting a peak appliance power that is higher than the aggregated one).

Due to the fact that the optimization problem models only the active state of the appliance, the algorithm is used to refine the output of the neural network only where an activation of the appliance is detected. The activation window is computed by looking at the neural network output. Each activation starts when a consumption greater than a value ACTSTART, and ends when the power consumption remains below ACTSTART for at least ACTTOLERANCE seconds. Both the values for ACTSTART and ACTTOLERANCE can be selected by looking at the ground truth power consumption over time of the appliance in the training set.

We can define the difference between the aggregated and predicted consumption as:

$$\Delta m_j = \int_{t_{j,start}}^{t_{j,end}} |P_j(t) - f_j(t) - \text{BASELINE}| dt \quad (8)$$

Where BASELINE is the average value of the mains power consumption before and after the current activation. The BASELINE offset is necessary due to the fact that in each instant there could be different sets of other appliances draining power, thus resulting in an unpredictable baseline.

The optimization problem takes into account both the predicted and aggregated data. As in training, it computes a cost that is used in the optimization objective. When it predicts an activation (and its corresponding $f_j(t)$), this cost is:

$$activec = w_c \Delta c_j + w_m \Delta m_j \quad (9)$$

where w_c and w_m are two appliance dependent hyperparameters that weights the contributions of the two costs.

Given that the refinement is triggered by the prediction of the neural network, the model must also consider the hypothesis that the neural prediction is a false positive. In this case, the role of the problem is to discard the prediction. Therefore, the optimization cost is computed as the power consumption predicted by the neural network.

$$inactivec = \int_{t_{j,start}}^{t_{j,end}} P_j(t)dt \tag{10}$$

Assuming that the parameters $\alpha_{j,i}^T, \beta_{j,i}^T, t_{j,i}^T, \Delta s^T, \Delta\alpha^T, \Delta\beta^T$ are the ones learnt during training, the optimization problem is defined as:

$$\begin{aligned}
&\underset{\alpha_{j,i},\beta_{j,i},t_{j,i}}{\text{minimize}} \ (\texttt{isactive})\ activec + (\overline{\texttt{isactive}})\ inactivec \\
&\text{subject to} \\
&\qquad t_{\min}^T d_B - \Delta s^T d < t_{j,i} < t_{\max}^T d_B + \Delta s^T d \\
&\qquad \alpha_{\min}^T - \Delta\alpha^T < \alpha_{j,i} < \alpha_{\max}^T + \Delta\alpha^T \\
&\qquad \beta_{\min}^T h_B - \Delta\beta^T h < \beta_{j,i} < \beta_{\max}^T h_B + \Delta\beta^T h \\
&\text{where} \\
&\quad t_{\min}^T = \min(\{t_{j,i}^T\}_{j=1}^m) \quad t_{\max}^T = \max(\{t_{j,i}^T\}_{j=1}^m) \\
&\quad \alpha_{\min}^T = \min(\{\alpha_{j,i}^T\}_{j=1}^m) \quad \alpha_{\max}^T = \max(\{\alpha_{j,i}^T\}_{j=1}^m) \\
&\quad \beta_{\min}^T = \min(\{\beta_{j,i}^T\}_{j=1}^m) \quad \beta_{\max}^T = \max(\{\beta_{j,i}^T\}_{j=1}^m)
\end{aligned} \tag{11}$$

Where `isactive` is a boolean variable that is true if the optimization problem predicts an activation, and false otherwise. The hyperparameters d_B, d, h_B, h are used to scale the values of the parameters $t_{j,i}, \beta_{j,i}$. The parameters $\alpha_{j,i}$ are not scaled to avoid losing the overall shape of the power consumption curve.

5 Experimental setting

This section describes the experimental setting that we defined to validate our proposed approach. In detail, we compare the predictions of our neuro-symbolic approach with respect to a fully neural approach. As a neural baseline, we use the model described in [15] which is also used as input to our neuro-symbolic approach. All the experiments have been run on the UK Domestic Appliance Level Electricity (UK-DALE) dataset which is one of the most used datasets in the literature to evaluate the performance of a disaggregation algorithm.

5.1 UK-DALE

UK-DALE contains the measurements of the energy consumption for the whole house and individual appliances of five UK houses. The readings have been collected by sampling every 6 seconds and refer to the period 11/09/2012-04/26/2017[4]. Each house hosts at least two occupants, with occupants be potentially different for type (family or not family) or habits (e.g. working all day). Therefore, the consumptions are not (always) the same for each house. More than 15 types of appliances are contained in the dataset but not all of them are in all houses. As done by other works like [15, 25], we focus on kettle, microwave, fridge, dishwasher and washing machine because these are the appliances having the highest impact on the total aggregated consumption.

[4] we used UK-DALE-2017

5.2 Seen setting

We evaluate our neuro-symbolic approach and the neural baseline on the "seen" scenario. Roughly speaking, it consists in seeing how both approaches behave when they have to predict the appliances' consumption over a period of time that they have not seen during training. In detail, given a time window $w^h = [s_w^h, e_w^h]$, $s_w^h, e_w^h \in \mathbb{N}$ with $s_w^h < e_w^h$, for an house h, we define two windows, w_{train}^h and w_{test}^h, where:

$$\begin{aligned} w_{train}^h &= [s_{w_{train}}^h, e_{w_{train}}^h] \\ w_{test}^h &= [s_{w_{test}}^h, e_{w_{test}}^h] \end{aligned}$$

with:

$$s_w^h \le s_{w_{train}}^h < e_{w_{train}}^h < s_{w_{test}}^h < e_{w_{test}}^h \le e_w^h$$

and we train and test both approaches using the data over the time windows w_{train}^h and w_{test}^h, respectively.

5.3 Metric

As done in other works like [15, 20, 25], we evaluate the predictions using the mean absolute error (MAE):

$$\text{MAE}(\hat{Y}_j, Y_j) = \frac{1}{l}\sum_{i=1}^{l} |\hat{y}_{ji} - y_{ji}|$$

where l denotes the length of the sequence (i.e. the length of the main) and $\hat{Y}_j$ and Y_j represent the predicted and the truth consumption sequence for appliance j (with $\hat{y}_{ji}$ and y_{ji} representing the consumption at time i).

6 Results

In table 1 are reported the MAEs for both the neural baseline and our neuro-symbolic approach on all the appliances. As can be seen by looking at the table, our neuro-symbolic approach outperforms the neural network on two out five appliances, while having comparable results on the remaining ones. In figure 4 are shown the prediction of both the neural network and our neuro-symbolic approach. On the left, is shown the comparison between the ground truth (blue) and the neural network (orange), while on the right is shown the comparison between the ground truth and our neuro-symbolic approach (green). As can be seen by looking at the first two rows (i.e, dishwasher and kettle), the use of background knowledge provided by our neuro-symbolic approach is useful to the neural network when the network captures the underlying consumption pattern, but it is not able to completely fill the gap with respect to the ground-truth. More precisely, our neuro-symbolic approach works when the predictions of the neural network are already quite accurate but are not consistent over time. This is not surprising since a trained appliance's network reasons in terms of time points, while our neuro symbolic approach reasons in terms of intervals of time (see section 4). For the remaining appliances, the lack of an improvement is due to different reasons: the objective of the constrained optimization problem consists in refining the predictions of the network, if they are already optimal (fridge) or too wrong (microwave) then the impact of the background knowledge is almost null; the training set does not contain enough representatives samples to model correctly the behaviour of the appliance. This is the case of the washing machine appliance that has not so many samples/activations and whose data contains a lot of variability. As a consequence, its corresponding MAE is low even though the predictions are not good.

In all the experiments, the automata model is learnt from (a subset of) data of the target house and then is always available. If not, we can use the training data of the other houses, but this would led to a drop in terms of performance, since we are going to learn an automata on a potentially different models of the same appliance which is not exactly what we want to predict. Furthermore, even though we consider only five appliances, the overall model can still be applied when there are more appliances. Indeed, despite having much more noise, which is due to the presence of more appliances that switch on and off, respectively, the model reasons on each appliance independently, and then it should work exactly the same.

Appliance	MAE nn	MAE neuro-symbolic
dishwasher	17.7	**8.02**
kettle	7.25	**4.52**
fridge	15.8	15.8
microwave	7.92	7.98
washingmachine	8.55	8.93

Table 1: MAE over the test set.

7 Conclusion and Future Work

In this paper, we propose a neuro-symbolic approach for NILM where background knowledge about the behaviour of a house's appliances is used on top of a neural network to refine its predictions. The refinement step is done through a (learnt) automata that change the prediction of the neural network according to the state's (exponential) function that has been learnt from the data (i.e., from a subset of activation windows). Experiments show that when the prediction of the neural networks are already quite accurate (i.e. the network is able to learn the pattern of consumption of an appliance) but are not consistent over time, the use of the background knowledge is able to ensure a greater consistency leading to a drastic increase of the performance. While these results are promising, there are several directions that can be investigated for future works. One consists in a more in-depth search for the parameters of the CSP. Indeed, we notice that a correct setting of the values of those parameters is fundamental in order to achieve better results. Therefore, an extensive evaluation can be conducted in order to set their values. Currently, one of the drawbacks of the datasets used in NILM is the availability of only few activations for each appliance making their generalization more difficult. Collecting more activations from different houses may increase drastically both the variability and generalizability of our proposed approach. Another direction, which could be also be conducted in parallel with the previous point, would be to move the setting from “seen" to “unseen", and see how our neuro-symbolic approach behaves when the prediction has to be done on a house (i.e., on its appliances) that it has not seen during training.

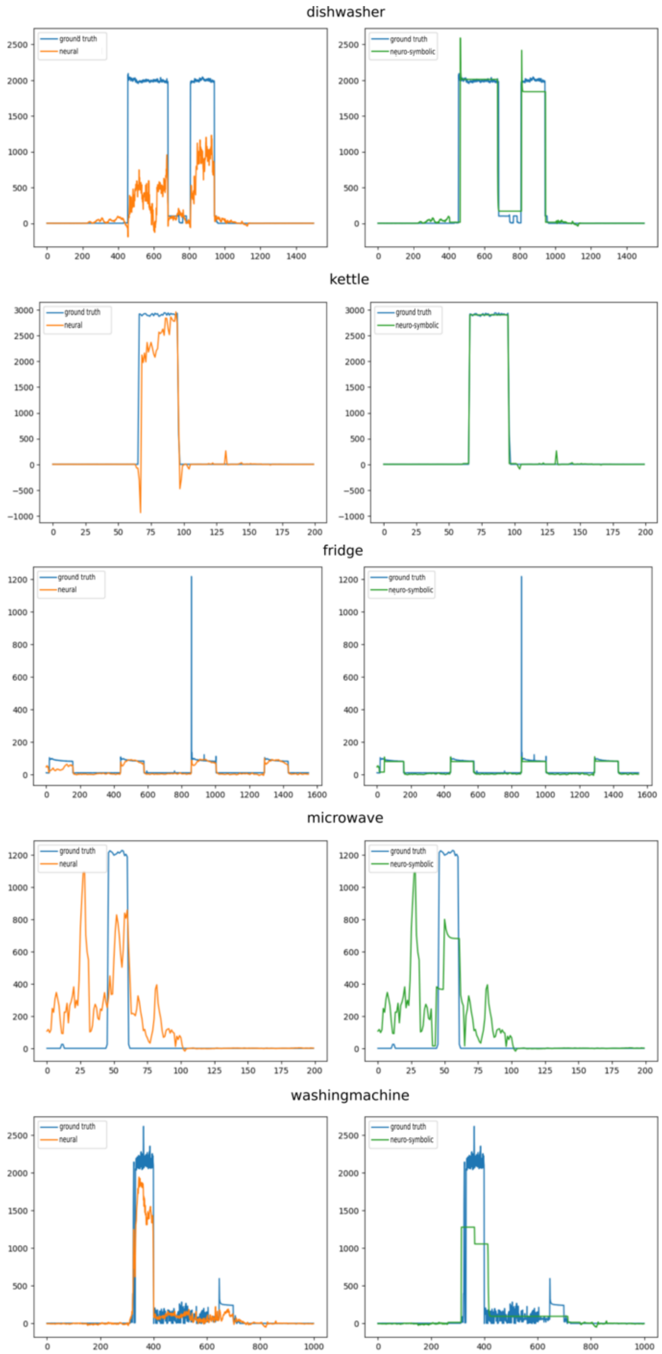

Figure 4: Prediction of the consumption for dishwasher, kettle, fridge, microwave, and washingmachine. Ground truth is in blue, while neural and neuro-symbolic are in orange and green, respectively.

References

[1] Gianluca Apriceno, Andrea Passerini, and Luciano Serafini, ‘A neuro-symbolic approach to structured event recognition’, in *28th International Symposium on Temporal Representation and Reasoning, TIME 2021, September 27-29, 2021, Klagenfurt, Austria*, eds., Carlo Combi, Johann Eder, and Mark Reynolds, volume 206 of *LIPIcs*, pp. 11:1–11:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, (2021).

[2] Gianluca Apriceno, Andrea Passerini, and Luciano Serafini, ‘A neuro-symbolic approach for real-world event recognition from weak supervision’, in *29th International Symposium on Temporal Representation and Reasoning, TIME 2022, November 7-9, 2022, Virtual Conference*, eds., Alexander Artikis, Roberto Posenato, and Stefano Tonetta, volume 247 of *LIPIcs*, pp. 12:1–12:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, (2022).

[3] Samy Badreddine, Artur d’Avila Garcez, Luciano Serafini, and Michael Spranger, ‘Logic tensor networks’, *Artif. Intell.*, **303**, 103649, (2022).

[4] Marco Balletti, Veronica Piccialli, and Antonio M Sudoso, 'Mixed-integer nonlinear programming for state-based non-intrusive load monitoring', *IEEE Transactions on Smart Grid*, **13**(4), 3301–3314, (2022).

[5] Marco Balletti, Veronica Piccialli, and Antonio Maria Sudoso, 'Mixed-integer nonlinear programming for state-based non-intrusive load monitoring', *IEEE Trans. Smart Grid*, **13**(4), 3301–3314, (2022).

[6] Junyi Chai, Hao Zeng, Anming Li, and Eric W.T. Ngai, 'Deep learning in computer vision: A critical review of emerging techniques and application scenarios', *Machine Learning with Applications*, **6**, 100134, (2021).

[7] Dong Chen, David Irwin, and Prashant Shenoy, 'Smartsim: A device-accurate smart home simulator for energy analytics', in *2016 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 686–692. IEEE, (2016).

[8] Ivan Donadello, Luciano Serafini, and Artur S. d'Avila Garcez, 'Logic tensor networks for semantic image interpretation', in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, ed., Carles Sierra, pp. 1596–1602. ijcai.org, (2017).

[9] Paolo Dragone, Stefano Teso, and Andrea Passerini, 'Neuro-symbolic constraint programming for structured prediction', in *Proceedings of the 15th International Workshop on Neural-Symbolic Learning and Reasoning as part of the 1st International Joint Conference on Learning & Reasoning (IJCLR 2021), Virtual conference, October 25-27, 2021*, eds., Artur S. d'Avila Garcez and Ernesto Jiménez-Ruiz, volume 2986 of *CEUR Workshop Proceedings*, pp. 6–14. CEUR-WS.org, (2021).

[10] Corinna Fischer, 'Feedback on household electricity consumption: A tool for saving energy?', *Energy Efficiency*, **1**, 79–104, (02 2008).

[11] R Gopinath, Mukesh Kumar, C Prakash Chandra Joshua, and Kota Srinivas, 'Energy management using non-intrusive load monitoring techniques–state-of-the-art and future research directions', *Sustainable Cities and Society*, **62**, 102411, (2020).

[12] G.W. Hart, 'Nonintrusive appliance load monitoring', *Proceedings of the IEEE*, **80**(12), 1870–1891, (1992).

[13] Yassine Himeur, Abdullah Alsalemi, Faycal Bensaali, Abbes Amira, and Ayman Al-Kababji, 'Recent trends of smart nonintrusive load monitoring in buildings: A review, open challenges, and future directions', *International Journal of Intelligent Systems*, **1**, (03 2022).

[14] *Neuro-Symbolic Artificial Intelligence: The State of the Art*, eds., Pascal Hitzler and Md. Kamruzzaman Sarker, volume 342 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2021.

[15] Jack Kelly and William Knottenbelt, 'Neural nilm: Deep neural networks applied to energy disaggregation', in *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*, BuildSys '15, p. 55–64, New York, NY, USA, (2015). Association for Computing Machinery.

[16] Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc D. Raedt, 'DeepProbLog: Neural Probabilistic Logic Programming', in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, eds., Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, volume 31, pp. 3753–3763, (2018).

[17] Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, and Marco Gori, 'LYRICS: A General Interface Layer to Integrate Logic Inference and deep Learning', in *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2019, Würzburg, Germany, September 16-20, 2019, Proceedings, Part II*, eds., Ulf Brefeld, Élisa Fromont, Andreas Hotho, Arno J. Knobbe, Marloes H. Maathuis, and Céline Robardet, volume 11907 of *Lecture Notes in Computer Science*, pp. 283—-298, (2019).

[18] Christoforos Nalmpantis, Nikolaos Virtsionis Gkalinikis, and Dimitris Vrakas, 'Neural fourier energy disaggregation', *Sensors*, **22**(2), 473, (2022).

[19] Daniel W. Otter, Julian R. Medina, and Jugal K. Kalita, 'A survey of the usages of deep learning for natural language processing', *IEEE Transactions on Neural Networks and Learning Systems*, **32**(2), 604–624, (2021).

[20] Veronica Piccialli and Antonio M. Sudoso, 'Improving non-intrusive load disaggregation through an attention-based deep neural network', *Energies*, **14**(4), (2021).

[21] Roberto Sebastiani and Silvia Tomasi, 'Optimization Modulo Theories with Linear Rational Costs', *ACM Transactions on Computational Logics*, **16**(2), (2015).

[22] Marc Roig Vilamala, Tianwei Xing, Harrison Taylor, Luis Garcia, Mani B. Srivastava, Lance M. Kaplan, Alun D. Preece, Angelika Kimmig, and Federico Cerutti, 'Using deepproblog to perform complex event processing on an audio stream', *CoRR*, **abs/2110.08090**, (2021).

[23] Zhun Yang, Adam Ishay, and Joohyung Lee, 'NeurASP: Embracing Neural Networks into Answer Set Programming', in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, ed., Christian Bessiere, pp. 1755–1762. ijcai.org, (2020).

[24] Michael Zeifman and Kurt Roth. Nonintrusive appliance load monitoring: Review and outlook, 2011.

[25] Chaoyun Zhang, Mingjun Zhong, Zongzuo Wang, Nigel Goddard, and Charles Sutton, 'Sequence-to-point learning with neural networks for non-intrusive load monitoring', in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, (2018).