# Chapter 5
# DNN Based Approach


**Abstract** The recent success of Deep Neural Networks (DNN) in several application scenarios drove the scientific community to employ this paradigm also for NILM. Kelly and Knottenbelt compared three alternative DNNs: in the first, they employed a convolutional layer followed by long short-term memory (LSTM) layers to estimate the disaggregated signal from the aggregate one. In the second, a denoising autoencoder composed of convolutional and fully connected layers is trained to provide a denoised signal from the aggregate one. The third network estimates the start time, the end time and the mean power demand of each appliance. The algorithms were evaluated on the UK-DALE dataset and showed superior performance with respect to the combinatorial optimization and FHMM algorithms implemented in the Non-intrusive Load Monitoring Toolkit (NILMTK).

**Keywords** Deep neural network · Denoising autoencoder · Footprint · Active power · Reactive power

## 5.1 Neural NILM

The work by Kelly and Knottenbelt [31] compared three different neural network architectures: in the first, they employed a convolutional layer followed by LSTM layers [60] to estimate the disaggregated signal from the aggregated one. In the second, a denoising autoencoder (dAE) composed of convolutional and fully connected layers is trained to provide a denoised signal from the aggregated one. The third network estimates the start time the end time, and the mean power demand of each appliance. The algorithms were evaluated on the UK-DALE dataset and the results showed that the dAE approach outperforms the alternative neural networks architectures as well as the FHMM algorithm implemented in the Non-intrusive Load Monitoring Toolkit (NILMTK) [73].



R. Bonfigli, S. Squartini, *Machine Learning Approaches to Non-Intrusive Load Monitoring*, SpringerBriefs in Energy,
https://doi.org/10.1007/978-3-030-30782-0_5

## 5.2 Denoising AutoEncoder Approach

The NILM task can be formulated as a denoising problem by expressing the aggregated signal as the sum of the power consumption of the appliance of interest and a noise component that incorporates all the remaining contributions. In particular, Eq. (2.1) can be reformulated as:

$$\overline{y}(t) = y^{(j)}(t) + v^{(j)}(t), \tag{5.1}$$

for $j = 1, 2, \ldots, N$, where

$$v^{(j)}(t) = \sum_{\substack{i=1 \\ i \neq j}}^{N} y^{(i)}(t) + e(t), \tag{5.2}$$

represents an overall noise term for the appliance $j$ that comprises both the measurement noise and the contributions of the other appliances. Thus, for obtaining $y^{(j)}(t)$, it would be sufficient to remove the noise term $v^{(j)}(t)$ from the aggregate measurement $\overline{y}(t)$.

In [31] and similarly in [30], noise removal is performed by means of a dAE, i.e., a neural network that is trained to reconstruct a clean signal from its noisy version presented at the input. Denoising autoencoders have been originally formulated in the context of *representation learning* and as an unsupervised training method [97]. The same structure has been later employed to perform actual noise removal, such as in speech related tasks [98, 99]. An autoencoder can be seen as an encoder network followed by a decoder network. The encoder provides an internal representation of the input signal and the decoder transforms it back into the input signal domain. A common choice consists in creating a network with specular encoder and decoder topologies. In the context of NILM, for each appliance, an autoencoder is trained to reconstruct the ground truth $y^{(j)}(t)$ given the aggregated signal $\overline{y}(t)$.

## 5.3 Algorithm Improvements

In this section, several algorithmic and architecture improvements to the dAE approach for NILM are proposed and an exhaustive comparative evaluation with the AFAMAP (Additive Factorial Approximate Maximum a Posteriori) algorithm [21] is conducted. In particular, compared to [31] the dAE approach for load disaggregation is improved by conducting a detailed study on the topology of the network, and by introducing pooling and upsampling hidden layers, and the rectifier linear unit (ReLU) activation function [100] in the output layer. Additionally, the

network output is recombined by using a median filter on the overlapped portions of the disaggregated signal. The second contribution is an exhaustive performance comparison between AFAMAP and the dAE approach. Indeed, FHMMs have been largely employed in the last years since they are an effective approach for load disaggregation, and AFAMAP, in particular, received noteworthy attention by the scientific community [101, 102], as described in Sect. 4.1. However, an exhaustive performance comparison between the two methods has not been yet conducted. Indeed, the authors of [31] compare their proposed approaches to the FHMM method implemented in NILMTK [73], but their comparison does not consider more advanced FHMM algorithms such as AFAMAP [21]. Additionally, their experiments consider only a noised scenario on a single dataset (UK-DALE). Here, the evaluation is performed on three datasets, UK-DALE [61], AMPds [58] and REDD [29] in different conditions: firstly, the algorithms are evaluated on denoised and noised scenarios. In the denoised scenario, the aggregated signal is the sum of the power profiles of the appliances that are disaggregated. In the noised scenario, the aggregated signal comprises also measurement noise and the contributions of unknown appliances. Successively, the algorithms generalization capabilities are evaluated by performing disaggregation on the data acquired in a house not considered in the training phase (unseen scenario). The performance is evaluated by using both energy-based metrics and state-based metrics [73]: the first, evaluate the capability of the algorithm to estimate the actual power profile of the appliances, while the second the capability of estimating whether the appliance is in the "on" or "off" state. In order to perform the experiments in presence of noise, a Rest-of-the-World (RoW) model has been introduced in the original AFAMAP [21] algorithm. This model represents all the appliances but the ones of interest and makes AFAMAP able to operate in a noised scenario. The obtained results show that on average the dAE approach outperforms AFAMAP in all the addressed experimental conditions.

The general network topology proposed here for NILM is shown in Fig. 5.1: the encoder network (Fig. 5.1a) is composed of one or more one-dimensional convolutional layers that process the input signal and produce a set of feature maps. Each convolutional layer is followed by a linear activation function, by a max pooling layer, and by additional convolutional and pooling layers. Finally, one or more fully connected layers followed by a ReLU [100] activation function close the encoder network. The max pooling operation returns the maximum value within a neighbourhood, and in image processing, it makes the obtained representation invariant to small translations of the input. In NILM, this translates into being more independent on the location of an activation inside an analysis window. Additionally, max pooling reduces the size of the feature maps and the number of units in the fully connected layers, thus reducing the number of training parameters. The ReLU activation function calculates the maximum between its input and zero, and in this case it prevents the occurrence of negative values of the disaggregated active power. The decoder (Fig. 5.1b) is structured specularly to the encoder, with

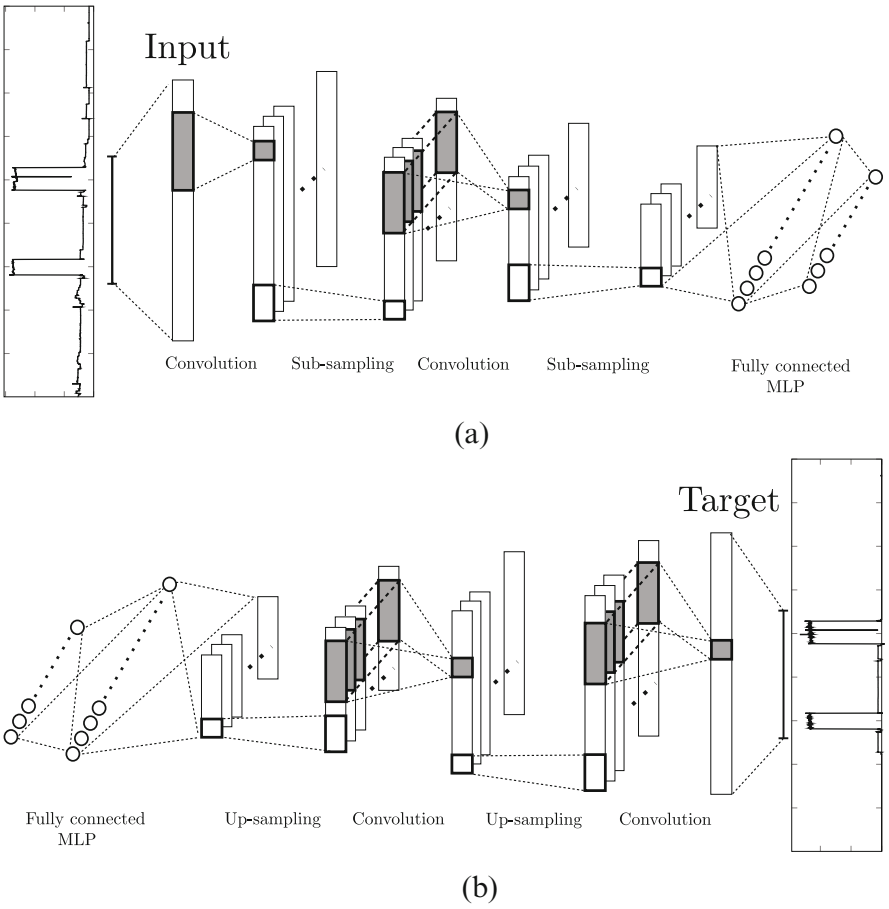


**Fig. 5.1** Generic autoencoder architecture employed for disaggregation. (**a**) Encoder network. The input signal is the aggregated power consumption. (**b**) Decoder network. The target signal is ground truth power consumption of each appliance

upsampling layers taking the place of max pooling layers. Compared to [31], several network topologies are explored, with multiple convolutional stages, max pooling and upsampling layers are introduced, and the ReLU activation function in the fully connected layers.

The dAE network is trained to minimize the mean squared error between its output and the activation of a single appliance. Training is performed by using the Stochastic Gradient Descent (SGD) algorithm with Nesterov momentum [103], and with the early-stopping criterion to prevent overfitting. The input data and the target are normalized in order to improve the learning efficiency.With respect to the reference work [31], several advancements have been introduced in the training phase. In particular, during the training phase, the initial value of the learning rate

is decreased when the performance on a validation set decreases. When this occurs, training is resumed from the epoch where the performance started decreasing. If the validation performance remains confined in a certain interval, typically when the learning process has reached the convergence or the learning rate has become too little, the early-stopping criterion is used. This is adopted in order to prevent overfitting.

In the disaggregation phase, the input signal $\overline{y}(t)$ is analysed by using sliding windows whose lengths depend on the size of the appliance activations. Windows are partially overlapped and the output signal is recombined by using a median filter on the overlapped portions. This differs from what proposed in [31], where the authors recompose the overlapped portions by calculating their mean value. The problem with this solution is that when an activation is only partially comprised in the analysis window, the network tends to underestimate the value of the output signal. As the window slides, the estimate increases, but averaging the overlapped portions produces an overall underestimated signal. Differently, by using the median operation on the overlapped portions, this phenomenon is mitigated, since greater values are preserved. The overall operation is depicted in Fig. 5.2.

The input signal is normalized following the same technique used in the training phase, while the disaggregated traces are denormalized after recombining outputs.

### *5.3.1 Experimental Setup*

In order to conduct an exhaustive evaluation on different scenarios, three public datasets have been chosen. The Almanac of Minutely Power dataset (AMPds) [58] contains recordings of consumption profiles belonging to a single home in Canada for a period of 2 years, at 1 min sampling period. The experiments are conducted by using six appliances: dryer, washing machine, dishwasher, fridge, electric oven and heat pump. The second dataset, UK-DALE [61], is composed of consumption profiles recorded in five houses in UK over 2 years, at 6 s sampling period. The houses consumptions are not equally distributed over this time period, e.g., house 3 contains only the kettle consumptions and some minor appliances recordings, thus it is not considered in the experiments. The five target appliances considered in all the experiments are: fridge, washing machine, dish washer, kettle and microwave. The third dataset, REDD [29], contains aggregate and circuit-level power profiles of several US households. The sampling period of the aggregate data is 1 s, while the one of the target profiles is 3 s, thus aggregate data was downsampled in order to match the sample period of the target profiles. The experiments are conducted by using four appliances: dryer, dishwasher, fridge and microwave. In the seen scenario, the data from two houses is used both for training and testing. In the unseen scenario, the same data is used for training, while testing is performed on the data of a third house.

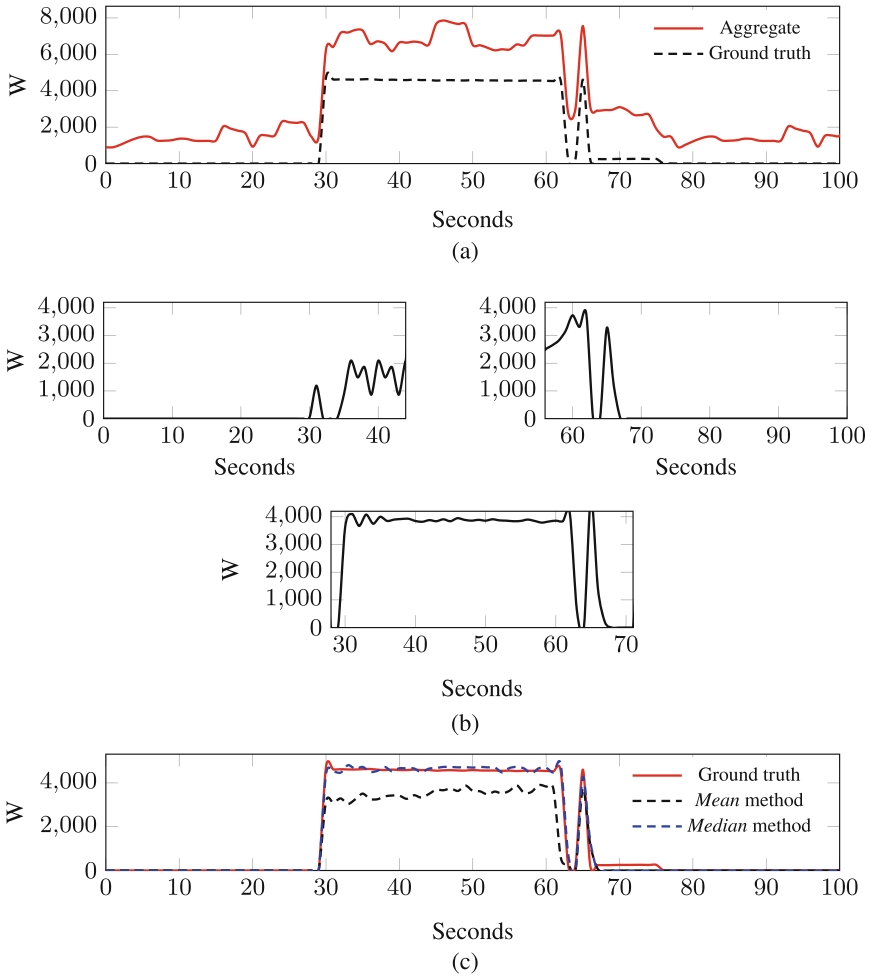


**Fig. 5.2** Network outputs recombined by using the mean operation and the median operation recombination on the overlapped portions. (**a**) A portion of aggregated data, analysed with sliding window technique. (**b**) Output of the dAE for each window. (**c**) Disaggregated traces comparison between *median* and *mean* recombining methods

The chosen appliances represent the principal contributions to the peak of power consumption in the aggregated signal, which allows us to consider the *denoised* scenario as an approximation of the *noised* scenario in the traits of higher power consumption. On the other hand, the *noise* contribution, assigned to the RoW model, depends on the number of remaining appliances not modelled and on the total energy of the main aggregated signal, and this affects the disaggregation performance in the

*noised* scenario. The *energy ratio* (ER), defined as:

$$\mathrm{ER} = \frac{E_{\mathrm{RoW}}}{E_{\mathrm{main}}} = \frac{\sum_{t=1}^{T} e(t)}{\sum_{t=1}^{T} \overline{y}(t)}, \tag{5.3}$$

expresses the energy proportion between the RoW model and the total aggregated data, and the values for each house in the considered datasets are showed in Table 5.1.

The datasets are split in different portions for training and testing, and their dimensions depend on the availability of appliances activations within the dataset. Regarding the training procedure, within the period specified in Table 5.11, the first 20% of activations are used to compose the validation set, while the remaining 80% are used for the models training (Table 5.2).

Regarding the ground truth consumption availability, two different scenarios can be defined. In the *seen* scenario, the disaggregation is computed on the same houses used to train the models, but in different period from the training data. In this scenario, both models, HMM and neural network, are created exploiting the same portion of training, in order to conduct a fair comparison between the methods. On the other hand, in the *unseen* scenario, the disaggregation is computed on the data related to a house not considered in the training phase. In this scenario, the ground truth consumptions related to each appliance are not available in the house where the disaggregation is performed, therefore no training data can be considered to create the models. The generalization property of the neural network allows to avoid a training procedure and to use the model trained on a set of data different from the test, whereas the footprints need to be suitably extracted in order to train the HMM. One possible approach, in this sense, is represented by the user-aided

**Table 5.1** Energy ratio (ER) for each house in the considered datasets

| | | UK-DALE | | | | REDD | | |
|---|---|---|---|---|---|---|---|---|
| Dataset | AMPds | House 1 | House 2 | House 4 | House 5 | House 1 | House 2 | House 3 |
| ER | 0.731 | 0.680 | 0.564 | 0.867 | 0.833 | 0.634 | 0.463 | 0.613 |

**Table 5.2** Definition of the training, validation and test sets for the considered datasets

| Dataset | Train+Validation | Test |
|---|---|---|
| AMPds | 1 year, 6 months | 6 months |
| *UK-DALE* | | |
| House 1 | 1 year, 8 months, 3 days | 7 days |
| House 2 | 4 months, 3 days | 7 days |
| House 4 | 6 months, 25 days | 7 days |
| House 5 | 2 months, 3 days | 6 days |
| *REDD* | | |
| House 1 | 33 days | 3 days |
| House 2 | 12 days | 2 days |
| House 3 | 12 days | 6 days |

footprint extraction algorithm, described in Sect. 4.4, that describes a procedure for the extraction of an approximated version of the appliance activations within the aggregated data when all the appliances are turned off, except the always-on in the house, i.e., the fridge and the freezer.

The experiments on the UK-DALE dataset have been performed as in [31], both for the *seen* and the *unseen* scenario.

The parameters related to the AFAMAP algorithm are defined as follows: the frame size is set to 60 min, which is an interval sufficiently large to include the whole activation for most of the appliances under study. For the ones with a longer activation, this frame size allows to include a complete operating subcycle, for which the HMM is still representative. The variance parameters are set to $\sigma_1^2 = \sigma_2^2 = 0.01$ according to the variance of the experimental data, and the regularization parameter is set to $\lambda = 1$. Table 5.3 presents the number of states, defined a-priori for each class of appliance. In the *denoised* scenario no parameters optimization has been conducted, whereas in the *noised* scenario, the number of the RoW states has been varied between the values {6, 8, 10} for both datasets.

The algorithm has been implemented in Matlab, and the CPLEX[1] solver has been adopted to solve the QP problem. The experiments have been conducted on a working station equipped with an Intel i7 CPU at 3.3 GHz, and 32 GB RAM. The time required for an experiments depends on the number of samples and the number of states of the HMM models: because of the different sampling rate between the datasets, the experiments last from 1 h for AMPds to 3 h for UK-DALE, while the introduction of the RoW model increases the simulation time up to 2 h for AMPds and 5 h for UK-DALE.

The parameters related to the dAE approach are defined as follows: each network receives data in a mini-batch of 64 sequences, and a mean and variance normalization is computed on the input data. In order to guarantee the same normalization over the whole dataset, the mean and variance values are computed from a random sample of the training set, whereas on the target data a min-max normalization is performed using the maximum power consumption value of the related appliance. The training data is composed of 50% of actual appliance related data, and 50% of synthetic data obtained by randomly combining real appliance activations. The training sequences have been extracted by using NILMTK [73]: this toolkit provides the method for the power activation extraction from the ground truth power consumption related to each appliance from both datasets. The data analysing window of the dAE needs to be enough large to comprise an entire activation of the

**Table 5.3** Number of states $m$ related to each class of appliance

| Nr. of states | Dryer | Washing machine | Dishwasher | Fridge | Electric oven | Heat pump | Kettle | Microwave |
|---|---|---|---|---|---|---|---|---|
| $m$ | 3 | 4 | 3 | 2 | 3 | 3 | 2 | 2 |

[1] https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/.

**Table 5.4** Window width (in samples) for the dAE architecture

| Dataset | Dryer | Washing machine | Dishwasher | Fridge | Electric oven | Heat pump | Kettle | Microwave |
|---|---|---|---|---|---|---|---|---|
| UK-DALE | – | 1024 | 1536 | 512 | – | – | 128 | 288 |
| AMPds | 75 | 120 | 210 | 45 | 120 | 90 | – | – |
| REDD | 1536 | – | 2304 | 496 | – | – | – | 96 |

The number of samples depends on the dataset sampling rate

appliance, but not too much to include other contributions, especially for appliances with short-duration activation. The window width depends on the appliance type, as described in Table 5.4.

As aforementioned, training has been performed by using the SGD algorithm with Nesterov momentum set 0.9. The maximum number of epochs has been set to 200 000, and the number of epochs for the variable step size technique has been set to 20 000. The initial value of the learning rate has been set to 0.1, with a decreasing factor equal to 10. The variable step size criterion has been applied on the $F_1^{(E)}$ calculated on the validation set, and the relative tolerance for early stopping criterion has been set equal to 0.01. The neural network has been implemented by means of the Lasagne library,[2] built on top of Theano [104]. All the network weights have been initialized randomly using Lasagne default initialization, without any layerwise pre-training.

In [31], the network topology is composed of an input and an output convolutional layer with 8 kernels of size 4. The middle layers consist of 3 fully connected layers with ReLU activation functions, where the number of neurons in the central layer is equal to 128, whereas for the other layers the number depends on the length of the input sequence. In the disaggregation phase, a hop size of 16 samples has been considered. The performance of this work represents the baseline for this approach. An intensive parameters optimization has been conducted regarding the number of kernels (N), size of each kernel (S), and the number of neurons in the central layer (H). The experiments have been conducted using each combination of parameters within the ranges: N={2, 4, 8, 16, 32, 64}, S={2, 4, 8, 16, 32, 64}, H={8, 16, 32, 64, 128, 256, 512, 1024, 2048}. Kernels larger than the input size have not been considered. The architecture that achieves the highest performance has been used as a starting point of an additional campaign of experiment, for which the first convolutional layer has been preserved, and a second stage, including pooling and up-sampling layers, has been introduced. The parameters have been varied within the same ranges defined above.

Max pooling is calculated on a segment with sizes equal to 2 or 4 samples, and the overlapped portion is either equal to half of the window or not present. For this new architecture the experiments have been conducted with a full search of the optimal parameters. The disaggregation phase has been carried out with a sliding

[2] https://lasagne.readthedocs.io/en/latest/.

window technique over the aggregated signal, using overlapped window with hop size in the range $\{1, 2, 4, 8, \frac{1}{4}window, \frac{1}{2}window\}$, where *window* represents the window width defined in Table 5.4.

The number of networks tested for each appliance in three datasets has been varied from 150 to 200, and this experimental campaign has been conducted on both *denoised* and *noised* scenario, in the *seen* and *unseen* conditions.

The experiments have been conducted on nVIDIA K80 GPUs. The training time varies depending on the network dimension and appliance type: because of the different sampling rates of the datasets, the experiments require from 2 to 10 h depending on the size of the training set.

### 5.3.2 Results

Regarding the AFAMAP algorithm, in the *noised* scenario, preliminary experiments have demonstrated that the highest performance is obtained when the number of states of the RoW model is 6. For the sake of conciseness, only the results for that number of states are reported.

For the same reason, the results of the entire experimental campaign of the dAE algorithm will not be reported. For each scenario, the introduction of the second stage of CNN improves the performance with respect to the single CNN stage for the majority of appliances, as well as the effectiveness of the pooling layer. The experiments demonstrated that a hop size with 1 and 2 samples results in the best performance.

For the AMPds and UK-DALE datasets, the dAE algorithm outperforms AFAMAP both in the *noised* and the *denoised* scenarios, as shown in Tables 5.5, 5.6, Fig. 5.5a, b. More in details, Fig. 5.5 shows the radar charts related to the $F_1^{(E)}$ metric for each appliance, and the area inside a line gives an overall performance indicator of the related approach. On the AMPds dataset, in the *denoised* case study, the absolute improvement in terms of $F_1^{(E)}$ amounts to $+17.3\%$, while in the *noised* scenario the absolute improvement amounts to $+13.3\%$. The same trend can be observed by considering the other metrics. Compared to AFAMAP, NEP reduces by 2.012 in the *denoised* scenario, whereas it reduces by 3.819 in the *noised* scenario. State-based metrics show a similar trend, since, in the *denoised* case study, $F_1^{(S)}$ improves by $+24.7\%$, while in the *noised* case study the absolute improvement is $+29.8\%$. Similar remarks apply to MCC. Analysing the performance of the individual appliances, the dAE algorithm outperforms AFAMAP for all the appliances in both the *denoised* and the *noised* scenario. In terms of $F_1^{(E)}$, the highest absolute improvement can be observed for the dishwasher ($+45.9\%$) in the *denoised* scenario, and for the oven in the *noised* scenario ($+48.4\%$). Considering the other metrics, the dAE algorithm outperforms AFAMAP for all the appliances in both scenarios, except for the fridge in the *noised* scenario, where AFAMAP achieves lower NEP and higher $F_1^{(S)}$. Indeed, for this appliance in the

**Table 5.5** Disaggregation performance in the seen scenario (AMPds dataset)

| Scenario | Algorithm | Metric | Dryer | Washing machine | Dishwasher | Fridge | Electric oven | Heat pump | Overall |
|---|---|---|---|---|---|---|---|---|---|
| Denoised | AFAMAP [21] | $F_1^{(E)}$ (%) | 87.3 | 14.5 | 44.4 | 35.5 | 38.1 | 76.9 | 60.4 |
| | | NEP | 0.281 | 7.761 | 2.093 | 0.837 | 2.909 | 0.352 | 2.372 |
| | | $F_1^{(S)}$ (%) | 60.7 | 7.4 | 11.9 | 36.0 | 5.0 | 86.2 | 50.3 |
| | | MCC | 0.631 | 0.092 | 0.161 | 0.335 | 0.121 | 0.855 | 0.366 |
| | dAE | $F_1^{(E)}$ (%) | 96.1 | 50.5 | 90.3 | 63.7 | 84.1 | 77.4 | **77.7** |
| | | NEP | 0.068 | 0.919 | 0.182 | 0.558 | 0.289 | 0.142 | **0.360** |
| | | $F_1^{(S)}$ (%) | 76.0 | 54.8 | 76.8 | 75.6 | 53.4 | 93.4 | **75.0** |
| | | MCC | 0.780 | 0.567 | 0.773 | 0.690 | 0.584 | 0.932 | **0.721** |
| Noised | AFAMAP + RoW | $F_1^{(E)}$ (%) | 65.3 | 6.2 | 27.8 | 38.3 | 8.9 | 54.6 | 40.8 |
| | | NEP | 0.999 | 18.100 | 2.812 | 0.938 | 6.305 | 0.873 | 5.004 |
| | | $F_1^{(S)}$ (%) | 16.0 | 7.6 | 10.7 | 43.3 | 2.0 | 55.5 | 30.8 |
| | | MCC | 0.239 | 0.096 | 0.141 | 0.198 | 0.041 | 0.543 | 0.210 |
| | dAE | $F_1^{(E)}$ (%) | 91.2 | 11.9 | 49.8 | 39.1 | 57.3 | 65.4 | **54.1** |
| | | NEP | 0.131 | 4.416 | 0.640 | 0.940 | 0.568 | 0.419 | **1.185** |
| | | $F_1^{(S)}$ (%) | 76.8 | 10.8 | 58.2 | 33.1 | 45.9 | 79.8 | **60.6** |
| | | MCC | 0.784 | 0.165 | 0.593 | 0.217 | 0.489 | 0.789 | **0.506** |

Numbers in bold indicate the best performing approach

**Table 5.6** Disaggregation performance in the seen scenario (UK-DALE dataset)

| Scenario | Algorithm | Metric | Kettle | Washing machine | Dishwasher | Fridge | Microwave | Overall |
|---|---|---|---|---|---|---|---|---|
| Denoised | AFAMAP [21] | $F_1^{(E)}$ (%) | 93.4 | 64.3 | 48.1 | 79.1 | 84.1 | 77.4 |
| | | NEP | 0.435 | 14.090 | 1.322 | 0.358 | 1.038 | 3.449 |
| | | $F_1^{(S)}$ (%) | 81.9 | 41.2 | 22.5 | 84.6 | 78.1 | 70.4 |
| | | MCC | 0.797 | 0.451 | 0.287 | 0.781 | 0.788 | 0.621 |
| | dAE | $F_1^{(E)}$ (%) | 94.1 | 59.6 | 86.2 | 85.8 | 82.9 | **81.8** |
| | | NEP | 0.087 | 13.087 | 0.220 | 0.207 | 0.287 | **2.777** |
| | | $F_1^{(S)}$ (%) | 95.7 | 56.2 | 57.4 | 93.2 | 90.4 | **82.1** |
| | | MCC | 0.957 | 0.559 | 0.620 | 0.896 | 0.903 | **0.787** |
| Noised | AFAMAP + RoW | $F_1^{(E)}$ (%) | 12.8 | 20.4 | 18.5 | 49.4 | 11.5 | 24.9 |
| | | NEP | 1.754 | 53.063 | 1.752 | 0.865 | 4.193 | 12.325 |
| | | $F_1^{(S)}$ (%) | 7.79 | 15.80 | 16.95 | 51.91 | 18.24 | 35.49 |
| | | MCC | 0.150 | 0.145 | 0.179 | 0.324 | 0.177 | 0.195 |
| | Kelly [31] | $F_1^{(E)}$ (%) | 80.1 | 35.1 | 58.2 | 64.1 | 59.5 | 60.4 |
| | | NEP | 0.522 | 1.384 | 0.707 | 0.609 | 0.923 | 0.829 |
| | | $F_1^{(S)}$ (%) | 82.12 | 35.32 | 69.53 | 65.68 | 62.58 | 69.18 |
| | | MCC | 0.821 | 0.372 | 0.706 | 0.575 | 0.626 | 0.620 |
| | dAE | $F_1^{(E)}$ (%) | 82.4 | 54.8 | 84.3 | 73.6 | 72.4 | **73.6** |
| | | NEP | 0.393 | 2.135 | 0.278 | 0.472 | 0.524 | **0.760** |
| | | $F_1^{(S)}$ (%) | 86.6 | 40.8 | 55.6 | 78.2 | 75.5 | **72.0** |
| | | MCC | 0.866 | 0.425 | 0.583 | 0.683 | 0.751 | **0.661** |

Numbers in bold indicate the best performing approach

noised scenario, the performance improvement in terms of $F_1^{(E)}$ is modest compared to the other appliances.

Compared to AFAMAP, in the UK-DALE dataset the absolute improvement in terms of $F_1^{(E)}$ is $+4.4\%$ in the *denoised* case study, and $+48.7\%$ in the *noised* scenario. The same trend can be observed by considering the other metrics: NEP reduces by 0.672 in the *denoised* scenario and by 11.564 in the *noised* scenario, while $F_1^{(S)}$ improves by $+11.7\%$ in the *denoised* case study and by $+36.51\%$ in the *noised* case study. MCC increases by 0.166 and by 0.466, respectively, in the *denoised* and in the *noised* scenario. Analysing the performance of the individual appliances, the dAE algorithm achieves superior performance for all the appliances in the *denoised* scenario, except for the washing machine and the microwave, for which the $F_1^{(E)}$ is similar. In the *noised* scenario, the dAE algorithm outperforms AFAMAP for all the appliances, with the highest improvement equal to $+69.6\%$ for the kettle. The same trend can be observed considering the other metrics. In the *noised* scenario, the optimization of the network parameters allows to outperform the dAE architecture presented in [31] for all the appliances, with the highest improvement of $F_1^{(E)}$ equal to $+26.1\%$ for the dishwasher. Considering the other metrics, the improvement follows the same trends, except for the washing machine evaluated in terms of NEP, and the dishwasher evaluated in terms of $F_1^{(S)}$ and MCC.

Regarding the REDD dataset (Table 5.7), in the *denoised* scenario the performance difference of the dAE algorithm with respect to AFAMAP varies with the evaluation metric. In particular, in terms of $F_1^{(E)}$ and MCC, AFAMAP outperforms the dAE algorithm, respectively, by 6.5% and 0.007. In terms of MCC, however, the relative improvement is limited, since it is equal to 0.95%. In terms of NEP and $F_1^{(S)}$, the dAE approach outperforms AFAMAP as shown in the experiments with the UK-DALE and AMPds datasets. This behaviour can be explained by considering that in the *denoised* seen scenario the HMM models in AFAMAP are trained by using data of the same building used in the disaggregation phase, while the network in the dAE approach is trained by using multiple buildings, and testing is performing on one of those. This aspect is less relevant in the *noised* scenario, because in AFAMAP the RoW model introduces a high variability in the disaggregation solution. Indeed, in this scenario the dAE approach outperforms AFAMAP regardless of the evaluation metric.

Generally, the dAE approach reaches higher disaggregation performance since it allows to reproduce complex activation profiles, which are learned during the training procedure and are associated to the aggregated profiles, even in the presence of the noise contribution. As shown in Tables 5.5, 5.6 and 5.7, the highest performance is reached in the disaggregation of the appliances with higher peak power consumption, since it allows a better association between the target and the aggregated input sequence during the training phase. In the HMM based approach, each state of an appliance model represents one value of power consumption, which does not allow to represent highly variable or transient phenomena between the working states of the appliance. Additionally, in the AFAMAP algorithm the disaggregation solution is obtained by considering all the appliance models at the

**Table 5.7** Disaggregation performance in the seen scenario (REDD dataset)

| Scenario | Algorithm | Metric | Dishwasher | Dryer | Fridge | Microwave | Overall |
|---|---|---|---|---|---|---|---|
| Denoised | AFAMAP [21] | $F_1^{(E)}$ (%) | 67.1 | 94.4 | 80.5 | 80.2 | **82.6** |
| | | NEP | 1.086 | 0.093 | 0.338 | 0.491 | 0.502 |
| | | $F_1^{(S)}$ (%) | 50.12 | 97.59 | 89.79 | 66.86 | 78.85 |
| | | MCC | 0.512 | 0.975 | 0.833 | 0.666 | **0.746** |
| | dAE | $F_1^{(E)}$ (%) | 66.3 | 87.3 | 74.6 | 64.9 | 76.1 |
| | | NEP | 0.515 | 0.265 | 0.543 | 0.397 | **0.430** |
| | | $F_1^{(S)}$ (%) | 71.7 | 92.7 | 80.5 | 69.6 | **80.9** |
| | | MCC | 0.669 | 0.926 | 0.666 | 0.695 | 0.739 |
| Noised | AFAMAP + RoW | $F_1^{(E)}$ (%) | 36.4 | 31.9 | 36.0 | 17.9 | 35.4 |
| | | NEP | 2.207 | 1.187 | 0.905 | 2.287 | 1.646 |
| | | $F_1^{(S)}$ (%) | 32.9 | 57.6 | 39.6 | 16.2 | 46.0 |
| | | MCC | 0.354 | 0.567 | 0.260 | 0.176 | 0.339 |
| | dAE | $F_1^{(E)}$ (%) | 62.1 | 72.5 | 66.9 | 53.0 | **66.1** |
| | | NEP | 0.551 | 0.506 | 0.760 | 0.615 | **0.608** |
| | | $F_1^{(S)}$ (%) | 64.0 | 81.8 | 70.9 | 61.6 | **72.4** |
| | | MCC | 0.495 | 0.814 | 0.468 | 0.604 | **0.595** |

Numbers in bold indicate the best performing approach

same time, while in the dAE approach each network operates independently from the others. This may cause a false energy assignment to an appliance, due to the need to satisfy the constraint that the sum of the reconstructed profiles corresponds to the aggregated power. In presence of noise, the performance degrades significantly, since the presence of the RoW, composed of a higher number of states compared to appliance models, increases the number of admissible solutions and, as a consequence, the chance of errors in the disaggregated profiles reconstruction. Moreover, in the AFAMAP algorithm there is no information on the total duration of the complete activation, since appliance models incorporate only the information on the working state transition and on the consumption values.

Further evaluations can be carried out by analysing the disaggregated profiles in *denoised* and *noised* scenario. Considering the UK-DALE experiments in *seen* scenario, the profiles related to the dishwasher in the house 1 are shown in Fig. 5.3. The appliance activation is correctly detected by the dAE in both scenarios, without producing false positives in the disaggregated trace. In the *noised* scenario, the reconstructed profiles have a high uncertainty, caused by the presence of noise in the aggregated power, but the average energy in the activation has a good correspondence with the ground truth one, which demonstrates the low degradation of performance compared to the *denoised* scenario. The same experiment has been considered for the fridge, whose profiles are shown in Fig. 5.4. The dAE algorithm recognizes the appliance activation in the *denoised* scenario, with a less accurate profile reconstruction in the activation overlapped with other appliances with respect to the isolated ones. Differently, the performance degrades in the *noised* scenario, with an incorrect activation detection and the production of some false positives, caused by the presence of noise in the aggregated signal.

As aforementioned, the *unseen* scenario is evaluated by using the UK-DALE and REDD datasets, due to the availability of recordings from several houses in both.

As in the *noised seen* scenario, preliminary experiments conducted by varying the number of states in the RoW model demonstrated that the highest $F_1^{(E)}$ is obtained with 6 states. Similarly, for the dAE algorithm the results of the entire experimental campaign will not be reported for the sake of conciseness. For each scenario, the introduction of the second stage of CNN and of the pooling operation improves the performance with respect to the single CNN stage for the majority of the appliances. Regarding the hop size in the sliding window disaggregation phase, as in the *seen* scenario the highest performance is reached by using 1 and 2 samples.

Similarly to the *seen* scenario in the UK-DALE dataset, the baseline [31] performance for each appliance in the *noised* scenario is outperformed by means of the optimization of the network parameters, with the highest absolute improvement of $F_1^{(E)}$ equal to $+30.2\%$ for the washing machine. The same trend can be observed for the other metrics, excepting for the $F_1^{(S)}$ and the MCC, where the dishwasher performance degrades.

For both datasets, the dAE algorithm outperforms AFAMAP in both scenarios, as shown in Tables 5.9 and 5.8. In the UK-DALE dataset, the absolute improvement in terms of $F_1^{(E)}$ amounts to $+8.6\%$ in the *denoised* case study, whereas it increases to

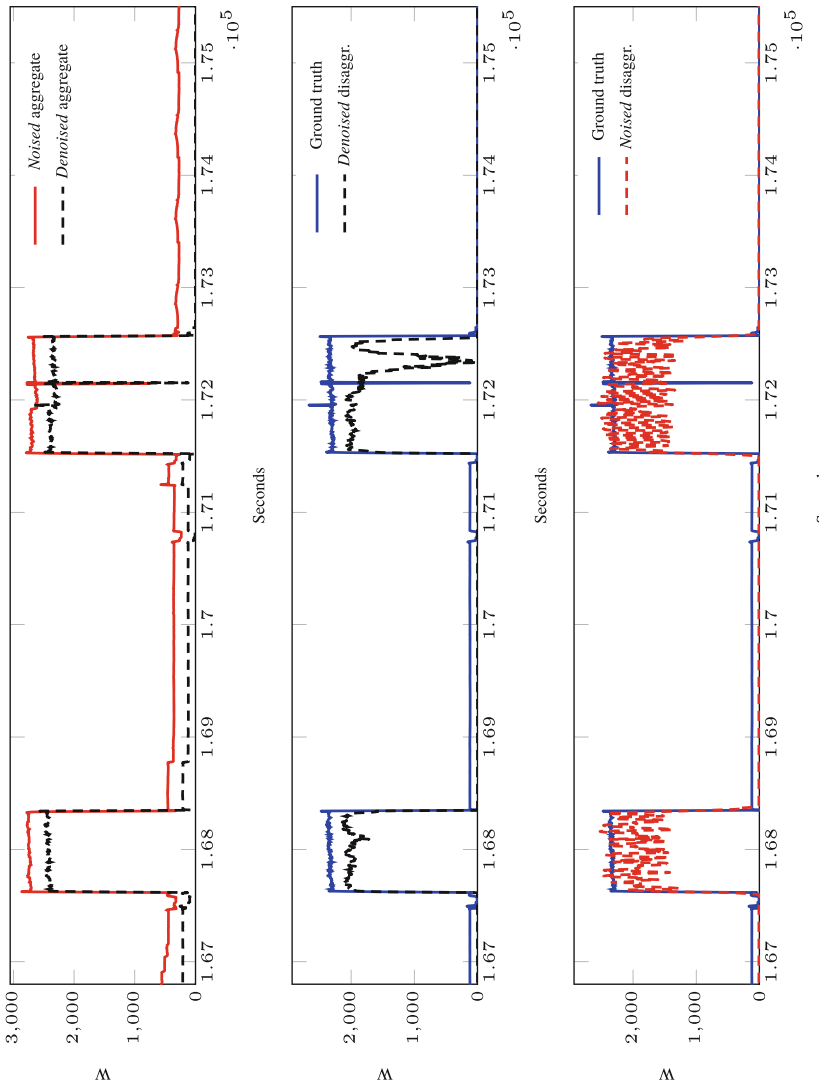


**Fig. 5.3** Disaggregated profiles in *denoised* and *noised* scenario in UK-DALE dataset, *seen* case study, related to the dishwasher in house 1

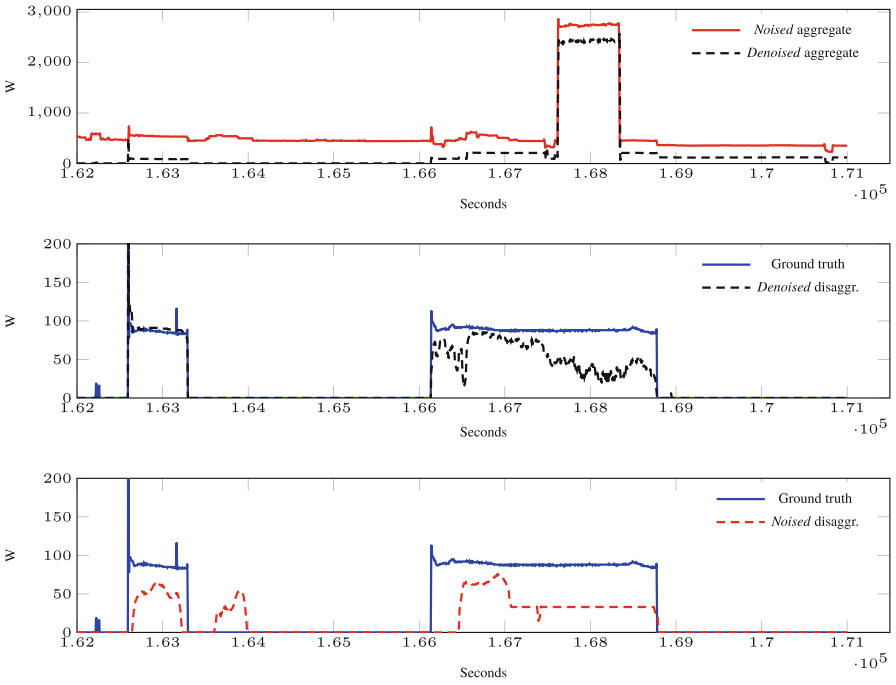


**Fig. 5.4** Disaggregated profiles in *denoised* and *noised* scenario in UK-DALE dataset, *seen* case study, related to the fridge in house 1

+ 50.5% in the *noised* scenario, demonstrating the superiority of the neural network based approach with respect to the HMM one, especially in presence of the noise contribution. The results evaluated with the other metrics confirm the same trend, with a reduction of NEP equal to 0.543 in the *denoised* case study and to 5.418 in the *noised* case study. Considering the state based metrics, the improvement evaluated with the $F_1^{(S)}$ amounts to + 12.52% in the *denoised* scenario and + 53.10% in the *noised*, as well as regarding the MCC with an absolute improvement of + 0.170 in the *denoised* scenario and + 0.594 in the *noised* scenario. As showed in Fig. 5.5c, overall the dAE algorithm outperforms AFAMAP both in the *denoised* and in the *noised* scenarios. In particular, the dAE exhibits a noteworthy robustness against the presence of noise, while the $F_1^{(E)}$ of AFAMAP reduces significantly. Observing the results of each appliance, the highest absolute improvement is obtained for the kettle and it is equal to + 80.4%. In the *denoised* scenario, the dAE algorithm outperforms AFAMAP for all the appliances, with the only exception of the dishwasher where the $F_1^{(E)}$ is 1.6% lower. Considering the other metrics, in the *noised* scenario, the performance is improved for all the appliances, while in the *denoised* scenario the same trend can be observed, except for the washing machine, which degrades its performance in terms of NEP, $F_1^{(S)}$ and MCC.

**Table 5.8** Disaggregation performance in the unseen scenario (REDD dataset)

| Scenario | Algorithm | Metric | Dishwasher | Dryer | Fridge | Microwave | Overall |
|---|---|---|---|---|---|---|---|
| Denoised | AFAMAP [21] | $F_1^{(E)}$ (%) | 32.2 | 49.5 | 69.3 | 7.0 | 46.4 |
| | | NEP | 3.336 | 0.811 | 0.491 | 4.754 | 2.348 |
| | | $F_1^{(S)}$ (%) | 18.6 | 89.8 | 73.6 | 4.3 | 55.9 |
| | | MCC | 0.282 | 0.901 | 0.650 | 0.056 | 0.472 |
| | dAE | $F_1^{(E)}$ (%) | 76.1 | 83.7 | 78.5 | 60.5 | **76.6** |
| | | NEP | 0.348 | 0.292 | 0.426 | 0.470 | **0.384** |
| | | $F_1^{(S)}$ (%) | 87.5 | 85.8 | 88.1 | 67.4 | **84.2** |
| | | MCC | 0.877 | 0.860 | 0.805 | 0.711 | **0.813** |
| Noised | AFAMAP + RoW | $F_1^{(E)}$ (%) | 22.5 | 18.8 | 40.0 | 8.4 | 26.4 |
| | | NEP | 3.803 | 1.521 | 0.946 | 3.728 | 2.500 |
| | | $F_1^{(S)}$ (%) | 14.2 | 41.3 | 37.3 | 5.1 | 35.0 |
| | | MCC | 0.228 | 0.399 | 0.180 | 0.083 | 0.222 |
| | dAE | $F_1^{(E)}$ (%) | 41.8 | 57.2 | 60.4 | 13.6 | **47.6** |
| | | NEP | 0.756 | 0.955 | 1.053 | 1.752 | **1.129** |
| | | $F_1^{(S)}$ (%) | 49.2 | 59.3 | 71.7 | 16.8 | **54.6** |
| | | MCC | 0.543 | 0.617 | 0.497 | 0.166 | **0.456** |

Numbers in bold indicate the best performing approach

**Table 5.9** Disaggregation performance in the unseen scenario (UK-DALE dataset)

| Scenario | Algorithm | Metric | Kettle | Washing machine | Dishwasher | Fridge | Microwave | Overall |
|---|---|---|---|---|---|---|---|---|
| Denoised | AFAMAP [21] | $F_1^{(E)}$ (%) | 95.1 | 32.3 | 80.9 | 73.6 | 29.9 | 66.1 |
| | | NEP | 0.114 | 2.089 | 0.457 | 0.449 | 3.311 | 1.284 |
| | | $F_1^{(S)}$ (%) | 97.11 | 25.59 | 12.84 | 74.68 | 38.33 | 61.68 |
| | | MCC | 0.971 | 0.353 | 0.177 | 0.690 | 0.440 | 0.526 |
| | dAE | $F_1^{(E)}$ (%) | 95.7 | 39.5 | 79.3 | 91.1 | 67.1 | **74.7** |
| | | NEP | 0.056 | 2.406 | 0.371 | 0.195 | 0.675 | **0.741** |
| | | $F_1^{(S)}$ (%) | 99.7 | 23.4 | 54.5 | 95.5 | 65.1 | **74.2** |
| | | MCC | 0.997 | 0.286 | 0.604 | 0.931 | 0.664 | **0.696** |
| Noised | AFAMAP + RoW | $F_1^{(E)}$ (%) | 3.2 | 4.7 | 20.2 | 32.2 | 4.2 | 17.3 |
| | | NEP | 3.087 | 6.559 | 2.078 | 1.021 | 18.413 | 6.231 |
| | | $F_1^{(S)}$ (%) | 0 | 5.1 | 11.8 | 33.6 | 3.3 | 18.7 |
| | | MCC | −0.001 | 0.085 | 0.151 | 0.120 | 0.090 | 0.089 |
| | Kelly [31] | $F_1^{(E)(\%)}$ | 79.1 | 23.3 | 39.2 | 65.1 | 20.6 | 50.8 |
| | | NEP | 0.448 | 1.607 | 0.892 | 0.562 | 2.875 | 1.277 |
| | | $F_1^{(S)}$ (%) | 93.9 | 26.5 | 55.9 | 77.8 | 30.9 | 66.8 |
| | | MCC | 0.940 | 0.373 | 0.597 | 0.712 | 0.416 | 0.608 |
| | dAE | $F_1^{(E)}$ (%) | 83.6 | 53.5 | 69.2 | 78.7 | 45.8 | **67.8** |
| | | NEP | 0.177 | 1.439 | 0.648 | 0.419 | 1.383 | **0.813** |
| | | $F_1^{(S)}$ (%) | 95.6 | 67.5 | 50.9 | 82.8 | 45.4 | **71.8** |
| | | MCC | 0.957 | 0.687 | 0.502 | 0.757 | 0.510 | **0.683** |

Numbers in bold indicate the best performing approach

On the REDD dataset, the absolute improvement in terms of $F_1^{(E)}$ amounts to $+30.20\%$ in the *denoised* scenario and $+21.18\%$ in the *noised* scenario. The other metrics follow the same trends, with a reduction of NEP equal to 1.964 in the *denoised* case study and to 1.371 in the *noised* case study. Considering the state based metrics, the improvement evaluated with the $F_1^{(S)}$ amounts to $+28.3\%$ in the *denoised* scenario and $+19.60\%$ in the *noised*, as well as regarding the MCC with an absolute improvement of $+0.341$ in the *denoised* scenario and $+0.234$ in the *noised* scenario. In the REDD dataset, differently from the *seen* scenario described above, the dAE algorithm outperforms on each appliance in both scenario, with the highest improvements in terms of $F_1^{(E)}$ of $+53.51\%$ for the microwave, except for the dryer in the *denoised* scenario with the state based metrics. The radar chart represented in Fig. 5.5e shows this improvement, and it represents the performance loss of both algorithm in the *noised* scenario with respect to the *denoised* scenario.

In the *unseen* scenario the generalization property of the dAE approach allows to apply the model without the need of training, with a reasonable degradation of performance. Regarding the AFAMAP algorithm, the approximation introduced by the footprint extraction procedure causes a lack of correspondence between the HMM and the appliance working states consumptions, and this results in a higher performance degradation, particularly in presence of *noise* where RoW model is present. This demonstrates the effectiveness of the neural networks approaches in an *unseen* scenario, which is the most interesting condition, because it represents a real-world application of the NILM service. As described in the previous section, the state based metrics confirm that the dAE produces a more reliable activation detection, with respect to the HMM based approach, even in an unseen scenario.

## 5.4 Exploitation of the Reactive Power

Besides machine learning techniques employed in order to solve the NILM problem, in the literature, neural networks (NNs) have been widely explored to address the problem of NILM.

Reactive power has already been identified as an exploitable feature to enhance NILM performances: starting from the seminal work of Hart [15], where the appliances working states are detected in the complex plan exploiting the active and reactive power consumption, up to the use of reactive power to train transient-state models [47, 50], In [105], the authors propose an active learning approach to significantly reduce the number of training samples needed to achieve high classification accuracies. In [106], the authors include reactive power trajectories, on top of which a principal component analyser is built to model each appliance. Finally, in [107] a recent approach based on finite-state machine modelling is built on top of real and reactive power signatures.

In this work the problem of NILM is addressed by using a particular family of NNs, that is the convolutional autoencoder. In particular, following the formalization

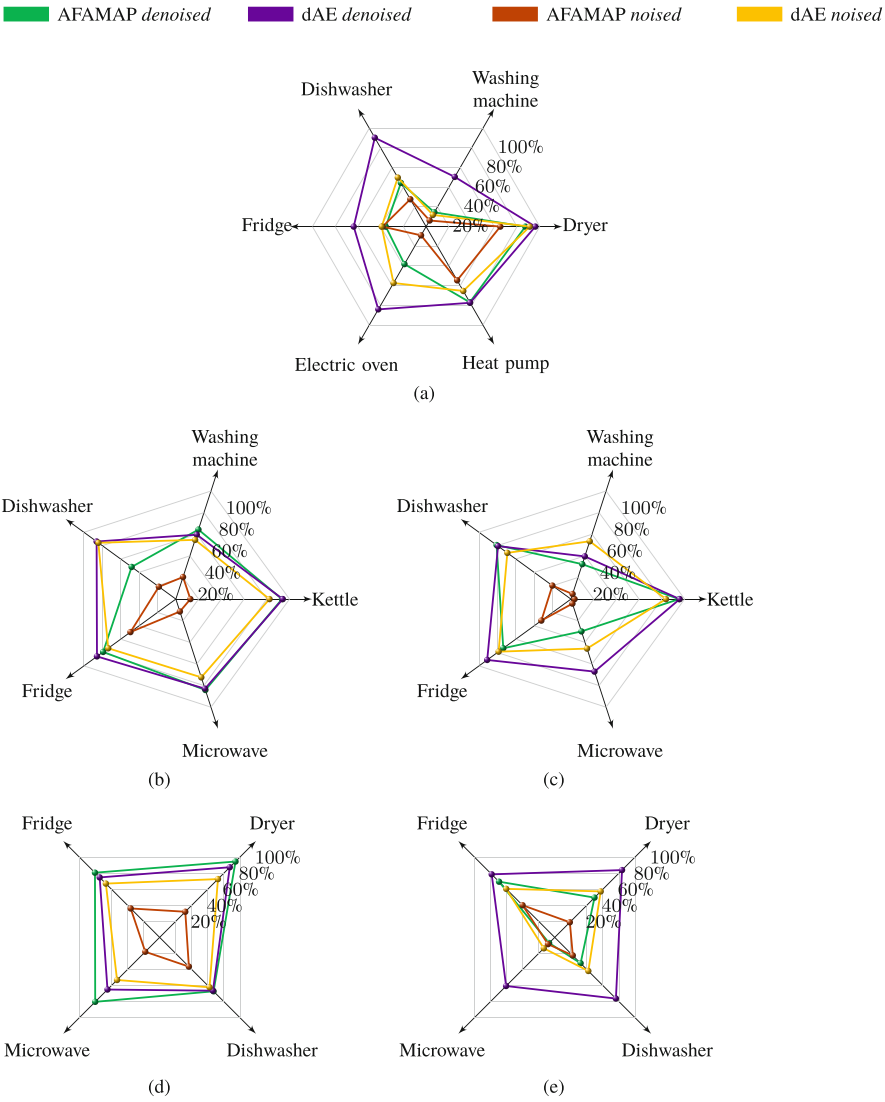


**Fig. 5.5** Performance for the different appliances for all the addressed algorithms. The $F_1^{(E)}$ (%) is represented. (**a**) Disaggregation performance on the AMPds dataset, *seen* scenario. (**b**) Disaggregation performance on the UK-DALE dataset, *seen* scenario. (**c**) Disaggregation performance on the UK-DALE dataset, *unseen* scenario. (**d**) Disaggregation performance on the REDD dataset, *seen* scenario. (**e**) Disaggregation performance on the REDD dataset, *unseen* scenario

of the problem as a denoising case study, the analysed architecture will be named *denoising autoencoder* (dAE) hereafter. As described in Sect. 5.2, denoising autoencoders architectures were deeply explored and several advancements were introduced, demonstrating that this approach reaches higher performance with respect to the FHMM based one [21].

In the majority of the methods discussed above, the signal under analysis in the disaggregation algorithm is represented by the active aggregate power consumption. The main focus of this work is the analysis on how the reactive power aggregate signal, used as input feature, influences the performance of dAEs. To do so, dAEs have been trained in an asymmetrical configuration, where the input consisted of both active and reactive aggregate power signals, and the output was solely the active power appliance trace. The proposed approach has been evaluated on two publicly available datasets, the *Almanac of Minutely Power* dataset (AMPds) [58] and the *UK Domestic Appliance-Level Electricity* (UK-DALE) [61] dataset. Despite not all appliances seem to benefit from the introduction of the reactive power feature, the overall averaged scores show significant improvements in all considered scenarios.

In the present examination, NILM has been formalized and treated as a denoising problem. In this scenario, the aggregate active power is seen as the superimposition of the most relevant appliance consumptions, plus a rest-of-the-world noise term, as described by the (2.1).

This equation highlights that, for each appliance, it is possible to retrieve the corresponding active power consumption $y_a^{(j)}(t)$ by removing the noise term from the whole aggregate signal.

The denoising problem stated above allows us to look at the dAE as a mapping function $f$ so that:

$$f : \mathbb{R}^{(L,1)} \Rightarrow \mathbb{R}^{(L,1)}, \tag{5.4}$$

where $L$ is the signal's window length. This means that the denoising function $f$ takes as argument a one-dimensional signal (the aggregate data) and retrieves, again, a one-dimensional vector: the disaggregated signal.

In introducing the reactive power signal, the active and reactive signals are concatenated on the second dimension, therefore the mapping function $f$ will now follows:

$$f : \mathbb{R}^{(L,2)} \Rightarrow \mathbb{R}^{(L,1)}. \tag{5.5}$$

This solution considers that the dAE will be driven to exploit the correlation existing between active and reactive consumptions. Invariance on the third axis is imposed by using one-dimension convolutional kernels, thus the active and reactive power signals are treated similarly to different colour channels in image processing tasks with convolutional neural networks.

Other configurations are possible, such as the concatenation on the time axis or the use of two separated dAE chains. These settings, however, would introduce

discontinuities in the input data, or, in the latter case, they would increase the computational complexity of the method. Due to these considerations, the proposed setting appeared as the best choice to make use of convolutional layers and to avoid excessive complications of the network topology.

In the present study, data has been pre-processed similarly to [31], where also the multiple values of $L$ (with reference to Eqs. 5.4 and 5.5) are given for each appliance. Hereafter, only a short description of the most salient pre-processing steps is given.

Firstly, for each appliance, active time windows are identified and grouped under the name of *activations*. An activation is defined as a time window in which the appliance consumption exceeds a *minimum active power* threshold for more than a *minimum ON time*. Moreover, if a subsequent consumption peak occurs before a pre-set *minimum OFF time*, it will be placed inside the same activation. Finally, all time windows between two adjacent activations will be grouped as *inactive sections*.

In constructing the target of an active sequence, one activation is randomly extracted and shifted: this way the network will be shown multiple perspectives of the same activation, making the most out of its informative potential. On the other hand, inactive sequence targets will be synthesized as zero-numbered vectors: its associated input will be an aggregate time window picked from the inactive section ensemble. The inactive input window will also be randomly shifted.

Finally, both the active and reactive components are standardized by subtracting the sequence mean value from each sample, and dividing it by the standard deviation calculated over the entire dataset:

$$\tilde{y}_c(t) = \frac{\overline{y}_c(t) - \overline{y}_{c,\text{mean}}}{\overline{y}_{c,\text{std}}}, \tag{5.6}$$

where $\overline{y}_{c,\text{mean}}$ is the active ($c = a$) and reactive ($c = r$) sequence mean value, calculated on $L$ samples, and $\overline{y}_{c,\text{std}}$ is the global standard deviation of the active and reactive signals. As mentioned in [31], this independent sequence centring does lose information, but it is able to improve the generalization capabilities of the network. Target sequences, on the other hand, are simply divided by the maximum power value of each appliance:

$$\underline{y}_a^{(i)}(t) = \frac{y_a^{(i)}(t)}{\max y_a^{(i)}(1:T)}, \tag{5.7}$$

where $\max y_a^{(i)}(1:T)$ is, for each appliance, the maximum power indicated in [31] and used in the activation extraction phase.

The dAE topology can be divided into two main stages: an encoder and a decoder. The first dAE stage, the encoder, takes as input the aggregate signal. The input is firstly processed by convolutional and pooling layers to extract shift-invariant features; then, fully-connected layers are used to extract higher-level feature representations. In the encoding phase, max-pooling is used as sub-sampling

function. The convolutional layers are composed exploiting linear activation function, whereas the fully-connected encoding layers are composed exploiting rectifier linear unit (ReLU). In this book, the encoder is composed of two convolutional layers and one fully-connected layer. The number of kernels and their size, the dimension of the max-pooling window and the number of units of the fully-connected layer have been explored in the experimental phase.

The encoder's output is fed to the decoder, whose hyper-parameter configuration and topology mirror the structure of the encoding network. Therefore, the decoder's input is firstly processed by fully-connected layers, followed by convolutional and up-sampling ones. The only noticeable difference between the encoder and the decoder topologies resides in the activation function: in the decoder, the rectifier function is used in place of the linear one. We remind that, despite generally being a symmetrical structure, the decoder output is always a uni-dimensional vector, meaning that, even when the reactive power is used as input, the output is always trained to match only the disaggregated active power signal.

Networks are trained with a supervised approach, aiming, for each input time window, to minimize the mean squared error between the disaggregated output and the (measured) corresponding appliance consumption. In order to minimize the mean squared error loss, the stochastic gradient descent algorithm is used, with the addition of Nesterov momentum [108] to further speed up the training convergence.

During training, networks are also shown synthetic sequences of data. The synthesis procedure is the same as described in [31], and it consists in randomly summing appliance activations with random shifts so to generate synthetic aggregate data. In addition to generating synthetic sequences, the algorithm will also make sure that active and inactive sequences will be used with a 50-50 ratio.

In order to prevent overfitting and excessive training times, an early-stopping criterion is used. However, in evaluating the model performance, the model's energy-based F1 score is used in place of the mean squared measure. Every time the model performance is checked on a validation set, the algorithm evaluates if an improvement has been made over the registered best score. If no improvements are encountered for a fixed number of training iterations, the difference between the last score and the best one is calculated; if such difference is higher than a certain threshold the learning rate is reduced and the training is re-started from the last best-performing configuration, otherwise the training is stopped. With such approach we aim at avoiding to stop the training when big score fluctuations occur (possibly) because the training cost function has not yet reached a stable minimum.

In the disaggregation phase, the whole aggregate signal is processed by the trained dAEs, which, for each appliance, reconstructs the corresponding consumption. The processing takes place with a sliding window approach, where overlapping sequences of fixed size are shown to the network, and the respective outputs are collected. In order to re-combine the overlapping sequences a median filtering has been used, since in Sect. 5.3 it was found to perform better than the average recombination used in [31].

Finally, at the end of the disaggregation phase, all samples are up-scaled by the same maximum power factor $\max y_a^{(i)}(1:T)$ previously used to scale the network targets.

### 5.4.1 Experimental Setup

AMPds contains recordings taken in a single house from 21 different power meters, with a sampling period of 60 s. The time period covered consists of 2 years, going from April 1, 2012 to March 31, 2014. Additionally to the active power consumption, AMPds also contains apparent and reactive power signals for the whole measurement period.

The UK-DALE dataset contains measurements taken in five different houses at multiple sampling rates. Differently from AMPds, in UK-DALE only active and apparent power measurements have been recorded, and this does not apply to all houses: in house three and four the aggregate active power signal was not measured. Therefore, in this evaluation only data taken from house one, two and five is used, with the sampling period set to 6 s.

Despite no reactive power measurements are available for the UK-DALE dataset, in order to retrieve the needed reactive power aggregate signals, the following relationship has been used:

$$\overline{y}_r(t) = \sqrt{\left(\overline{y}_{ap}(t)\right)^2 - \left(\overline{y}_a(t)\right)^2}, \tag{5.8}$$

where $\overline{y}_r(t)$, $\overline{y}_{ap}(t)$ and $\overline{y}_a(t)$ represent the reactive, the apparent and the active power sample in each sequence, respectively. On the AMPds, on the other hand, reactive power measurements allowed us to evaluate the magnitude of its contribution over the active power, at appliance level consumption. In particular, as shown in Table 5.10, reactive over active signal ratios were calculated for each appliance. What emerges is that the reactive power's magnitude oscillates between 7.6% and 26.6% of the active one, thus highlighting that, in this scenario, reactive power can indeed be considered a significant additional feature.

As shown in Table 5.11, data has been divided, for both datasets, into training, validation and test sets. In particular, after training activations are extracted, 20% of

**Table 5.10** Reactive over active (R/A) power ratios on the AMPds

| Appliance | R/A (%) |
|---|---|
| Dishwasher | 7.6 |
| Electric oven | 10.3 |
| Fridge | 9.5 |
| Heat pump | 20.0 |
| Tumble dryer | 16.1 |
| Washing machine | 26.6 |

**Table 5.11** Training, validation and test data subdivision by start and end date

| Dataset | Number of buildings | Train+Validation set | Test set |
|---|---|---|---|
| AMPds | 1 | 2012-10-01 | 2012-04-01 |
| | | 2014-04-01 | 2012-10-01 |
| UK-DALE | 1 | 2013-04-12 | 2014-10-22 |
| | | 2014-10-21 | 2014-12-15 |
| | 2 | 2013-05-22 | 2013-09-27 |
| | | 2013-09-26 | 2013-10-10 |
| | 5 | 2014-06-29 | 2014-09-01 |
| | | 2014-09-01 | 2014-09-07 |

**Table 5.12** Seen and unseen building subdivision for the UK-DALE dataset

| Appliance | Train/seen test | Unseen test |
|---|---|---|
| Dishwasher | 1, 2 | 5 |
| Fridge | 1, 2 | 5 |
| Kettle | 1, 2 | 5 |
| Microwave | 1, 2 | 5 |
| Washing machine | 1, 5 | 2 |

them is used to form validation batches, and the remaining 80% is used to construct train batches.

For each appliance trained on the UK-DALE dataset, data from one of the three available houses is excluded from training. This allows to define two different test cases, namely the *seen* and the *unseen* scenarios. By doing so, the aim is to test more deeply the networks' ability to generalize, since, in the unseen scenario, the model is not given the possibility to overfit the corresponding appliance signal. In Table 5.12 the seen/unseen house subdivision is reported.

Here a description of the parameter setup used in our experiments is given. Firstly, the window sizes used for each appliance are the same as shown in Sect. 5.3. These window sizes were identified as best-performing by Kelly and Knottenbelt in [31]; also all thresholds used during the activation extraction phase are the same as indicated in Kelly's article.

At the beginning of the training phase, network parameters are initialized with a random distribution: control is taken over this and all other random processes via the pre-setting of the code's random seed. The network training is conducted batch-by-batch, with a batch size fixed to 64 sequences, and the same size is used also for the validation batch. The maximum number of training iterations is fixed to 200,000, and the validation check is performed once every 10 iterations. We choose 2000 to be the maximum number of no-improvement iterations: when reached, the algorithm will decide whether to stop the training or to reduce the learning rate by a factor of 10.

In order to evaluate different network topologies, a grid search has been conducted on the encoder hyperparameters. All the combinations of the following

hyperparameters have been evaluated:

- first layer kernels: 32, 128;
- kernel window size: 4, 16, 32;
- pooling size: 2, 4;
- fully-connected layer size: 512, 4096.

In addition, the number of kernels in the second convolutional layer is double than the first layer ones, and the pooling and window sizes are equal in both convolutional layers. As aforementioned, the topology of the decoder mirrors the one of the encoders. Considering each appliance and the two datasets, the total number of experiments run is 528. Moreover, it has to be highlighted that, given the absence of reactive and apparent power measurements at appliance level for the UK-DALE, the data synthesis procedure described in Sect. 5.3 has not been possible. Therefore, only appliances trained on the AMPds made use of the described data augmentation technique. The activation extraction procedure explained in Sect. 5.3 has been performed by using the Non-Intrusive Load Monitoring Toolkit (NILMTK) [73]. The experimental framework (available upon request) has been developed in Python (v. 2.7.10) and Keras (v. 2.1.2) over the Theano [104] backend (v. 0.9.0). Finally, the hardware setup used to run our experiments were NVIDIA GeForce GTX 970 and TITAN X graphic processing units.

### *5.4.2 Results*

In Tables 5.14 and 5.13 experimental results obtained on both datasets are reported. Moreover, to better visualize score trends, the reader can refer to Fig. 5.6, where radar graphic representations of the scores are showed.

Observing AMPds results, it is possible to notice that all appliances benefit from the introduction of the reactive power input, the only exception being the electric oven. By looking at the overall scores, it is possible to identify an improvement of 8.1% if both the active and reactive aggregate signals are used instead of the active-only input.

**Table 5.13** F-score results (%) on the UK-DALE dataset

| | Seen | | Unseen | |
|---|---|---|---|---|
| Appliance | Active | Active + Reactive | Active | Active + Reactive |
| Dishwasher | 71.6 | **83.3** | 44.3 | **50.6** |
| Fridge | 68.5 | **70.8** | 68.9 | **76.7** |
| Kettle | 89.0 | **89.9** | **82.1** | 80.2 |
| Microwave | 64.4 | **80.7** | 37.0 | **67.9** |
| Washing machine | 35.5 | **49.8** | 5.4 | **23.3** |
| Overall | 67.7 | **76.1** | 58.0 | **62.9** |

Bold values represent the experiment with the higher result, therefore the best algorithm in the experiment comparison

**Table 5.14** F-score results (%) on the AMPds

| Appliance | Active | Active + Reactive |
|---|---|---|
| Dishwasher | 62.9 | **77.5** |
| Electric oven | **66.3** | 65.0 |
| Fridge | 37.4 | **43.4** |
| Heat pump | 72.7 | **76.3** |
| Tumble dryer | 94.8 | **95.5** |
| Washing machine | 34.3 | **59.5** |
| Overall | 62.1 | **70.2** |

Bold values represent the experiment with the higher result, therefore the best algorithm in the experiment comparison

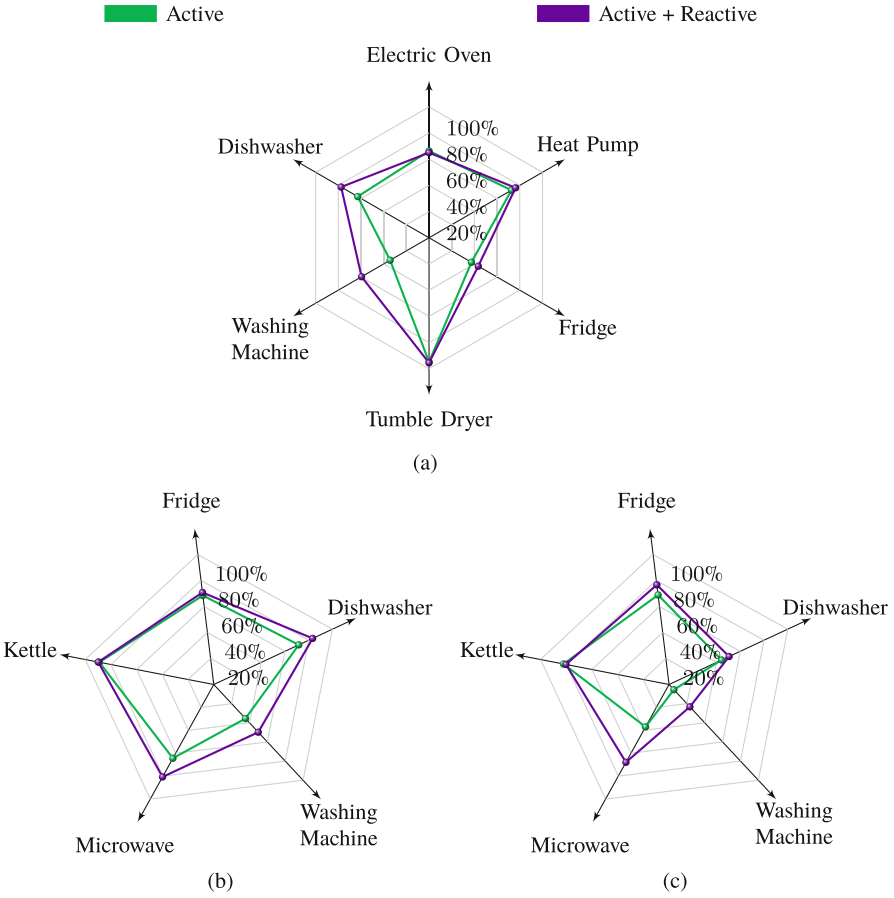


**Fig. 5.6** Performance for the different appliances for the all the addressed algorithms. The $F_1^{(E)}$ (%) is represented. (**a**) Disaggregation performance on the AMPds dataset, *seen* scenario. (**b**) Disaggregation performance on the UK-DALE dataset, *seen* scenario. (**c**) Disaggregation performance on the UK-DALE dataset, *unseen* scenario

Concerning the UK-DALE dataset, improvements can be observed on both the seen and the unseen scenarios. In particular, for the seen scenario, all appliances show score improvements ranging from 0.9% for the kettle, to 16.3% for the microwave, resulting in an overall improvement of 8.4%. On the unseen scenario, on the other hand, only the kettle showed reduced performance with the introduction of reactive power: −1.9%. The overall score, however, improves by 4.9%, highlighting that score increases still outweigh the reduction encountered.

Finally, it is worth highlighting that, given the nature of the proposed algorithm, an ensemble technique can be employed. As there is no dependence among appliance models, it is possible to choose the best-performing input configuration, namely active power only or active and reactive power. A possible strategy for choosing whether to use the reactive power as additional feature is by observing the F1 scores obtained in the validation phase. On the AMPds dataset, this solution translates into using the active power only model for the electric oven, and the active and reactive power models for the remaining appliances, resulting in a 0.2% improvement. This also shows that the validation score generally gives a reliable information on which configuration can be preferred. The ensemble technique has no effect on the UK-DALE dataset, since the validation scores obtained by using the active and reactive power models are always higher than the scores obtained with active power only models.