Attention in Recurrent Neural Networks for Energy Disaggregation

Virtsionis Gkalinikis Nikolaos, Christoforos Nalmpantis, and Dimitris Vrakas

School of Informatics, Aristotle University of Thessaloniki, Thessaloniki 54124, Greece
{**virtsion, christofn, dvrakas**}@csd.auth.gr
https://www.csd.auth.gr/en/

Abstract. Energy disaggregation refers to the separation of appliance-level data from an aggregate energy signal originated from a single-meter, without the use of any other device-specific sensors. Due to the fact that deep learning caught great attention in the last decade, numerous techniques using Artificial Neural Networks (ANN) have been developed to accomplish this task. Whereas most of the current research focuses on achieving better performance, the goal of this paper is to design a computationally light deep neural network based on attention mechanism. A thorough analysis shows how the proposed model is implemented and compares the performance of two different attention layers in the problem of energy disaggregation. The novel architecture achieves fast training and inference with minor performance trade-off when compared against other computationally expensive state-of-the-art models.

Keywords: energy disaggregation, non-intrusive load monitoring, artificial neural networks, attention

1 Introduction

Energy disaggregation provides the ability to estimate the electrical energy consumption of an appliance, using only the total power consumption of a house. It is also known as non-intrusive load monitoring (NILM). With the use of a disaggregation algorithm on the aggregate signal the power of the target device is approximated. Further analysis can identify inefficiencies of the various appliances, in order to reduce their energy usage. Additionally, with the use of NILM, the electrical energy management may be improved towards a direction of nullifying the unnecessary waste of energy usage, one of the crucial factors of climate change and global warming.

In modern times, smart meters are used in a more frequent fashion among residential houses and buildings [1], causing NILM to be one of the most trending energy data analytics techniques [2] in the residential and small commercial sector. Smart houses integrate home energy management systems (HEMS) in order to monitor and manage electrical appliances, reducing energy cost for consumers. In HEMS appliance load monitoring (ALM) can be achieved with either intrusive or non-intrusive monitoring methods [3]. The main advantage of NILM against intrusive-loading monitoring (ILM) is that it requires measurements from a single mains meter instead of multiple meters. Load monitoring is cheaper and more straightforward,

although ILM offers higher accuracy.

This paper contributes to the research of NILM in two major points. First, with the design of a lightweight model using artificial neural networks. As a result, faster training and inference times were achieved with a minor performance decrement in comparison to a state-of-the-art architecture. In the second place, with the introduction of Attention in the task of energy disaggregation alongside with promising results.

The structure of this article is as described bellow. To begin with, the related work about NILM and energy disaggregation is presented. Secondly, the idea and purpose of Attention is described alongside with the corresponding related work. Section 3.1 includes a short explanation of the calculations inside the attention mechanism. In section 4, there is an in depth analysis of the novel architecture and a presentation of its purpose and benefits. Next, the methodology of experiments is described. In section 6, there is a presentation of the most important results. Finally there are conclusions and proposals for future work.

2 Related Work

The problem of energy disaggregation states back to mid 1980s when it was firstly introduced by Hart. Hart [4] proposed a combinatorial optimization method in order to extract the optimal states of the target appliances so that the sum of power consumption would be the same as the meter reading. This method is applicable only on devices that have finite number of states, thus it cannot be used on appliances with variable consumption.

NILM research interest has raised a lot with the rise of internet of things. For a long time one of the most popular methods solving the energy disaggregation problem was Factorial Hidden Markov Models (FHMM), an extension to Hidden Markov Models (HMMs). In FHMM the architecture consists of multiple independent HMMs in parallel, where the observed output is a combination of all the hidden states. Kolter and Jaakkola used additive FHMMs, where the output was the sum of all the independent HMMs outputs [5].

The rise of machine learning and deep learning pushed researchers to use techniques from the sectors of Natural language processing (NLP), Computer Vision and Time Series Analysis. In 2015, Kelly and Knottenbelt [6] described three novel architectures using three different kinds of ANNs, an LSTM network, a denoising autoencoder architecture and a network to regress start/end time and power. These models outperformed Hart's algorithm and FHMM on experiments executed on the UK-DALE [7] data set. Mauch and Yang [8] investigated an other method using a recurrent network with LSTM neurons on low frequency (<1kHz) real power data. The experiments were executed on REDD [9] dataset alongside synthetic data. This approach showed good performance for appliances with recurring patterns allowing low frequency power measurements.

In 2017 Zhang et al. [10] implemented an architecture called Sequence-to-Point using CNNs layers, outperforming the results of Kelly and Knottenbelt [6]. A key point

of difference of this model in respect to the recurrent architectures in [6] and [8] is that a window of aggregate data is considered in order to predict the appliance consumption on a single time step, thus the name Sequence-to-Point. On the other hand, in [6] and [8] a single time step of the aggregate signal is used to predict the device power consumption at a the same time step. Krystalakos et al. [11] used Gated Recurrent Units (GRUs) instead of LSTMs alongside with dropout layers in order to improve the efficiency of previous RNN architectures. Furthermore, a sliding window approach was used in similar manner as proposed in [10], where the model receives a window of past data and predicts the power consumption at a single point.

Due to the lack of a benchmark method, the comparison of various methods and models most of the time is questionable. In an effort to efficiently tackle this challenge, Symeonidis et al. [12] proposed a set of experiments as a benchmark basis. In addition, the Stacking method of five popular architectures is explored resulting in promising results on 2-state devices. In a nutshell, regarding the matters of reproducibility and comparability of NILM frameworks, it is suggested to be a standardization of the assessment procedures [13,14].

Although there has been great progress in the energy disaggregation field using Deep Learning and ANNs, the deployment of NILM systems is still questionable. Due to the large number of parameters, these models have slow training and inference times. Additionally, NILM research mostly focuses on designing one model per device, resulting that a complete NILM system should intergrate as many models as the number of devices the target environment contains. Thus, these type of architectures are not directly applicable in real time situations, where energy measurements are obtained with high sampling frequencies providing huge quantities of data. The creation of lightweight architectures is a first and important step in order to achieve successful deployment of NILM on embedded systems. The next step is to consider multi-label machine learning models, where one model is trained in order to identify more than one appliances. Basu et al. [15,16] were the first to introduce the multi-label classification in NILM tasks, with the use of known machine learning algorithms such as decision trees and boosting. The most recent work considering multi label classification in energy disaggregation published by Nalmpantis and Vrakas [17], where a novel framework called multi-NILM is proposed. In multi-NILM approach, a dimensionality reduction technique called Signal2vec [18] is combined with a lightweight disaggregation model, achieving better results versus a state of the art multi-label classification approach.

3 Attention Mechanism

One of the most common tasks in machine learning is to extract input-output relations such as in machine translation and image captioning, where source and target sequences have different lengths in general. In Deep Learning, the most popular way of dealing with this format of tasks is with sequence to sequence models (seq2seq). The original seq2seq architecture (Sutskever et al. [19]) consists of

two essential RNNs; the encoder and the decoder. The encoder's role is to compress the sequential input into a context vector of fixed length, which contains a summary of the source sequence. On the other side, given the context vector, the decoder's purpose is to construct the target sequence. The calculation of the context vector is derived after processing each time step of the input and keeping the last hidden state of the encoder. In practice seq2seq models fail to process very long sequences. One possible reason can be attributed to the fixed length of the context vector.

The purpose of attention, as introduced by Bahdanau [20], is to assist the decoder to focus on the most important parts of the input. It provides information between the entire input sequence and the decoder output at each time step. The idea is that at every time step of the decoder an alignment vector is computed containing the score between the input's sequence and the decoder's output at the corresponding moment. As a result, the context vector is a combination of the alignment vector and the encoder's output. The model successfully focuses on the relevant parts of the input sequence.

There are different types of attention, depending on how scores and alignments are computed. The most common ones are the Additive [20] and the Multiplicative/Dot [21]. In addition, Cheng [22] proposed a different attention mechanism called Self-Attention which is also referred as intra-attention. The benefit is that different positions of the same inputs are related. Self-Attention can adopt both Bahdanau's and Luong's scoring functions. The neural network that is proposed in this research incorporates Additive and Dot attention mechanisms.

3.1 Dot and Additive Attention

In general an Attention layer receives three kind of vectors; query, key and value. Depending on the query, attention computes an output based on the key and value. The steps to calculate the output are described below and depicted in Fig. 1.

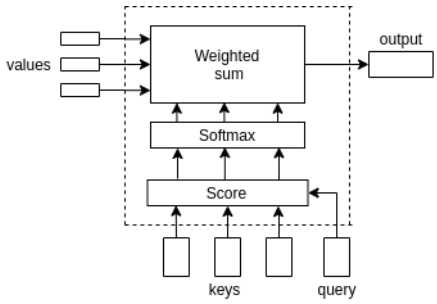

Fig. 1: Inside Attention mechanism.

Firstly, a score function is used to measure the similarity between a query (q) and a key (k_i) and for each query-key pair, scores (a_i) are computed.

$$a_i = score(q, k_i) \tag{1}$$

Secondly, these scores are normalized to add to one, using a softmax. Thus, the attention weights are obtained as follows.

$$b_i = \frac{\exp(a_i)}{\sum_j \exp(a_j)} \tag{2}$$

The last step is to combine the values (v) and the attention weights (b) as a weighted sum.

$$output = \sum_{i=1}^{n} b_i v_i \tag{3}$$

The main difference between Additive and Dot attention mechanisms is the scoring function. As the name suggests, in the Dot mechanism scores between keys and queries are computed by calculating the dot product. On the other hand, Additive attention computes scores as a non-linear sum.

For the purpose of using an attention layer between the CNN and GRU layers the idea of Self-Attention was used. In Self-Attention the aim is to learn the dependencies between all the parts of the same input sequence. In this set up as query, key and value inputs the Attention layer receives the output of the CNN layer.

4 Neural Network Architecture

The goal of this paper is to design a computationally light neural network. Being inspired by Window GRU (WGRU), a lightweight architecture has been developed. The novel model is called Self-Attentive-Energy-Disaggregation (SAED) and is up to 7.5 times faster in training and up to 6.5 times faster in inference, while there is trivial trade-off in performance.

WGRU consists of a CNN layer, two Bidirectional GRU layers and one Dense layer before the output layer. Also drop out was used to prevent overfitting. In order to design a less demanding model the most natural step is to try to reduce the number of parameters. However, we noticed that reducing the parameters was leading to dramatic performance decrease. Thus, the key concept was to find an alternative layer and GRU was replaced by the Attention Layer.

SAED combines the benefits of three different types of layers. Firstly a 1D convolution layer extracts new features. 1D convolution layers can recognize local patterns in a sequence at certain positions of a sequence, which can later be recognized at different positions. As a consequence, 1D convnets are time invariant. Next, the attention mechanism learns to focus on the most important features. Following a recurrent neural network is capable of extracting sequential patterns. Lastly, the dense layer acts as a regressor, giving the final result. The architecture is shown in Fig. 1.

It is important to point out that in the proposed model the Attention layer

functions as a Self-Attention mechanism receiving as input only the output of the CNN layer.

Recently, Tensorflow officially released two Attention layers, Attention layer which corresponds to Dot Attention and AdditiveAttention layer. In this paper the model comes with either Additive or Dot attention mechanism, mentioned as SAED-add and SAED-dot correspondingly.

As optimization algorithm Adam was used [23], while the loss was measured using mean squared error. The model was developed using Keras and Tensorflow 2.2.0 and all the experiments executed on a Nvidia GPU GTX-1060 6Gb. NILMTK framework [24] was used for loading and prepossessing the data.

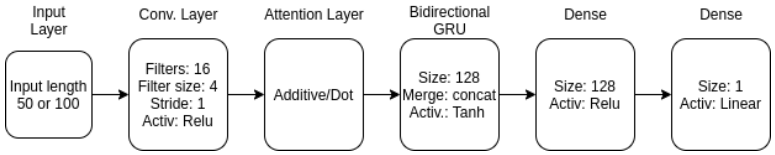

Fig. 2 Architecture of the Attention model.

5 Methodology of Experiments

For the experiments only real data was used with sampling period of 6 seconds and batch size of 1024. According to previous research [11], the optimal size of the input vector is device-dependent. The devices that were chosen alongside the sliding window sizes are presented in Table 1. All the models were trained for 5 epochs following the benchmark methodology described in [12], where the experiments are divided in four categories; Single Building NILM, Single Building learning and generalization on same dataset, Multi building learning and generalization on same dataset and Generalization to different dataset. The Experiments were executed following that order.

Table 1. Sliding window sizes (in samples) used for each device.

Device	WGRU	Dot Attention	Additive Attention
Dish Washer	50	50	50
Fridge	50	50	50
Kettle	50	50	50
Microwave	50	50	50
Washing Machine	100	100	100

The first category is about experiments where training and testing are applied on the same house at different time periods, in order to evaluate the model in the same environment where training took place. If a model doesn't perform well in this category of experiments it is probably weak [12]. The second category of experiments refers to training and inference on different buildings of the same data set. The purpose of these experiments is to inspect the generalization potential of

the model on unseen buildings. Different buildings mean that different appliances are used, the residents have different habits, resulting in divergent energy patterns. Within the same data set though, similarities of the energy footprint of each building are also expected. These similarities are mainly attributed to properties of the electricity grid, common seasonality or weather conditions and regionality. Thus, more experiments are required to evaluate the generalization ability of the model in depth.

In the third category, training data is collected from different houses of the same data set and inference is executed on an unseen building, while in the last category of experiments training data is also collected from different houses although the model is tested on houses of a different data set.

The fact that in these experiments the training data is composed from different houses, evaluates the sufficiency of the model in learning from multiple/different sources. In addition, the challenge for the model is higher in the last category, because it has to successfully learn from high variety data and infer on unseen data from a different data set. These two categories (especially the last) are considered as tough tasks for the models and if a model excels in them then it is considered very strong [12].

In this paper all the models were trained and tested using the UK-DALE [7] data for the first three categories, while as test data for the fourth category of experiments we used the REDD [9] data set. These data sets are considered dissimilar, because they are originated from different countries; UK-DALE contains measurements of house-hold devices in UK and REDD measurements of house-hold devices in USA.

All the experiments are summed up in Table 1. For Kettle the fourth category of experiments was not executed due to the lack of kettle device in the REDD data.

Table 2. Buildings used for train and inference.

	Category 1		Category 2		Category 3		Category 4	
Device	Train	Test	Train	Test	Train	Test	Train	Test
Dish Washer	1	1	1	2, 5	1,2	5	1,2	1,2,3,4,6
Fridge	1	1	1	2,4,5	1,2,4	5	1,2,4	1,2,3,5,6
Kettle	1	1	1	2,3,4,5	1,2,3,4	5	-	-
Microwave	1	1	1	2,3,5	1,2	5	1,2	1,2,3,5
Washing M.	1	1	1	2,4,5	1,5	5	1,5	1,2,3,4,5,6

For categories 1 and 2 the training on house 1 from UK-DALE is during the first 9 months of 2013 while the inference contains the last 3 months of the same year. For categories 3 and 4 the ratio of test versus training data depends on each device. In addition, for some devices the REDD data contains very few measurements which resulted in bad results even for the State of the art model.

In order to evaluate and compare the models with Attention versus the state of the art WGRU model, three metrics are used; F1 score, Relative Error in Total Energy (RETE) and Mean Absolute Error (MAE). The purpose of the F1 score is to evaluate

the ability of model to detect on/off energy states. MAE (measured in Watts) and RETE (dimensionless) are used in order to measure how capable is the model in predicting the actual electrical power consumed by the device.

Considering as E' the predicted total energy, E the true value of total energy, T the length of the predicted sequence, y_t' the inferred power consumption and y_t the true value of power consumption at time point t, the metrics are calculated as:

$$F1 = 2\frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

$$RETE = \frac{|E' - E|}{max(E', E)} \tag{5}$$

$$MAE = \frac{1}{T}\sum_{}^{} |y_t' - y_t| \tag{6}$$

Given the number of true on state predictions (TP), false on state predictions (FP) and false off state predictions (FN), Precision and Recall are computed as such:

$$Precision = \frac{TP}{TP + FN} \tag{7}$$

$$Recall = \frac{TP}{TP + FP} \tag{8}$$

6 Results and Comparisons

Due to the size of the results, the most important of them are presented in Tables 3 – 21, where the best are highlighted. Also, the average duration of a training epoch, measured in seconds GPU, is mentioned. The complete set of results alongside with the supplementary code are provided in the following github repository: https://github.com/Virtsionis/SelfAttentiveEnergyDisaggregator.

Table 3. Dish Washer, Category 1

Model	F1	RETE	MAE	epoch(s)
WGRU	**0.33**	**0.17**	13.22	550
SAED-dot	0.28	0.31	13.03	77
SAED-add	0.25	**0.17**	**12.03**	141

Table 4. Dish Washer, Category 2

Model	F1	RETE	MAE	epoch(s)
WGRU	0.26	0.77	37.47	550
SAED-dot	**0.63**	**0.62**	**33.48**	77
SAED-add	0.6	0.63	34.31	141

Table 5. Dish Washer, Category 3

Table 6. Dish Washer, Category 4

Model	F1	RETE	MAE	epoch(s)
WGRU	0.23	0.42	43.33	575
SAED-dot	0.25	0.46	**43.3**	74
SAED-add	**0.52**	**0.37**	44.48	138

Model	F1	RETE	MAE	epoch(s)
WGRU	0.39	0.6	34.35	575
SAED-dot	**0.41**	0.19	**27**	74
SAED-add	0.18	**0.13**	36.16	138

As shown in Table 3, in Category 1 experiments of Dish Washer the SAED models perform on par with WGRU in up to 7.1 times faster training time per epoch. In Category 2, SAED-dot is the clear winner with similar metric values as the SAED-add model, but with almost half the training time per epoch. In Category 3 of the same device, Table 5, the SAED models show better performance. Specifically the SAED-add performs better in respects to F1 and RETE, whereas in terms of MAE all the models perform the same. As presented in Table 6, in Category 4 the SAED-dot achieves better F1 score and MAE, while SAED-add has lower METE. It occurs that SAED shows promising results on Dish Washer in comparison to the WGRU, with faster training and better performance in Categories 2-4.

In similar manner, Tables 7-10 present the results on the Washing Machine for Categories 1-4 accordingly. In Category 1, SAED-dot is 7.5 times faster than WGRU trading of maximum 10% performance regarding the metrics F1 and MAE. As presented in Table 8, in Category 2 SAED-dot performs on par with WGRU but with 7.5 times faster training time per epoch. Results for Category 3 are shown in Table 9, where the SAED models have 7% greater F1 score than the WGRU and similar MAE. In terms of RETE in this category of experiments, the WGRU is a clear winner. Results of the fourth category of experiments can be found in Table 10. In this category, the SAED models are trained 7.2 times faster and with lower RETE and METE values than the WGRU.

It is notable that disaggregating Dish Washer and Washing Machine, the SAED models have comparable or better performance with the WGRU while training time per epoch was up to 7.5 times faster.

Table 7. Washing M., Category 1

Model	F1	RETE	MAE	epoch(s)
WGRU	**0.54**	**0.12**	**16.55**	1097
SAED-dot	0.51	0.26	18.51	147
SAED-add	0.45	0.29	28.55	416

Table 8. Washing M., Category 2

Model	F1	RETE	MAE	epoch(s)
WGRU	**0.34**	0.43	**10.45**	1097
SAED-dot	0.3	**0.34**	13.1	147
SAED-add	0.3	0.53	22.01	416

Table 9. Washing M., Category 3

Model	F1	RETE	MAE	epoch(s)
WGRU	0.52	**0.16**	**39.09**	585
SAED-dot	**0.56**	0.36	51.39	81
SAED-add	0.38	0.24	40.06	81

Table 10. Washing M., Category 4

Model	F1	RETE	MAE	epoch(s)
WGRU	**0.26**	0.66	43.65	585
SAED-dot	0.18	**0.39**	50.65	81
SAED-add	0.18	0.7	**41.93**	81

Results for the Fridge are summed in Tables 11-14. As presented in Table 11, in Category 1 WGRU achieves greater F1 score while SAED-add shows promising results with the smallest RETE and MAE of the three models, reaching up to 4 times faster training times. In Category 2, WGRU is a clear winner, whereas in Categories 3 and 4

the SAED models perform the same as the WGRU showing good generalization capabilities.

Table 11. Fridge, Category 1

Model	F1	RETE	MAE	epoch(s)
WGRU	**0.63**	0.268	33.29	562
SAED-dot	0.27	0.225	12.11	73
SAED-add	0.27	**0.131**	**10.94**	145

Table 12. Fridge, Category 2

Model	F1	RETE	**MAE**	epoch(s)
WGRU	**0.82**	**0.13**	**28.46**	562
SAED-dot	0.62	0.6	35.25	73
SAED-add	0.66	0.65	32.31	145

Table 13. Fridge, Category 3

Model	F1	RETE	MAE	epoch(s)
WGRU	**0.53**	0.32	**52.57**	519
SAED-dot	0.49	**0.29**	50.89	69
SAED-add	0.5	0.33	51.39	70

Table 14. Fridge, Category 4

Model	F1	RETE	MAE	epoch(s)
WGRU	**0.52**	**0.18**	**51.18**	519
SAED-dot	**0.52**	0.29	51.35	69
SAED-add	**0.52**	0.22	50.52	70

In Categories 1 and 2 of the Kettle, shown in Tables 15-16, the three models have comparable RETE and MAE values, but the WGRU achieves the best F1 score in 7.7 slower training time. In the third category of experiments presented in Table 17, the WGRU is the winner in terms of F1 and RETE, whereas in MAE all the models perform the same.

The above results reveal that the SAED models show difficulties in disaggregating two-state devices in comparison to the WGRU. Especially, in Categories 1-2 of the Fridge and the Kettle the SAED has low values on F1 score, but it achieves good results in Categories 3-4 of the Fridge. The low values of F1 score indicates the difficulty of the models to identify the On/Off states of the test devices.

Table 15. Kettle, Category 1

Model	F1	RETE	MAE	epoch(s)
WGRU	**0.657**	**0.09**	**7.35**	563
SAED-dot	0.442	0.14	8.58	73
SAED-add	0.335	0.26	9.46	143

Table 16. Kettle, Category 2

Model	F1	RETE	MAE	epoch(s)
WGRU	**0.9024**	0.311	**14.042**	563
SAED-dot	0.6233	0.302	19.035	73
SAED-add	0.4868	**0.278**	17.349	143

Table 17. Kettle, Category 3

Model	F1	RETE	MAE	epoch(s)
WGRU	**0.41**	**0.05**	**9.92**	1096
Dot Attention	0.273	0.27	12.2	141
Additive Attention	0.305	0.18	10.9	271

The results of the experiments on the Microwave are displayed in Tables 18-21. As presented in Tables 18-19, in Categories 1-2 the WGRU performs better than the SAED models in terms of F1. In the same categories, the SAED performs on par with the WGRU regarding the RETE and MAE metrics. In the third category of experiments SAED models outperform the WGRU, where in Category 4 WGRU achieves 17% better F1 score in 10 times slower training time. Considering that the Microwave is a multi-state device with variable power consumption and on-state duration, the SAED

models show descent performance comparing with the WGRU.

Table 18. Microwave, Category 1

Model	F1	RETE	MAE	epoch(s)
WGRU	**0.32**	**0.09**	**6.29**	560
Dot Attention	0.16	0.14	7.51	74
Additive Attention	0.18	0.16	7.61	144

Table 19. Microwave, Category 2

Model	F1	RETE	MAE	epoch(s)
WGRU	**0.44**	0.25	**4.36**	560
Dot Attention	0.25	0.19	5.97	74
Additive Attention	0.26	**0.17**	5.98	144

Table 20. Microwave, Category 3

Model	F1	RETE	MAE	epoch(s)
WGRU	0.08	0.59	60.53	440
Dot Attention	0.21	0.58	**56.93**	41
Additive Attention	**0.22**	**0.51**	59.36	41

Table 21. Microwave, Category 4

Model	F1	RETE	MAE	epoch(s)
WGRU	**0.41**	0.2	**23.53**	440
Dot Attention	0.34	0.2	25.67	41
Additive Attention	0.34	**0.15**	25.13	41

Overall, the SAED models achieve good performance in disaggregating multi-state devices instead of successfully detecting two state devices. Furthermore, the SAED performs good in experiments of Categories 3-4, a fact that reveals the great generalization capability of the proposed models.

An important and frequently neglected parameter when comparing models is the inference time. It is obvious that the size of test data affects the duration of inference. In order to compare the models, the inference time for several sizes of test data was measured. The inference time of each model, when disaggregating, is presented in Fig. 3, where 1 day of data is equal to 14351 samples.

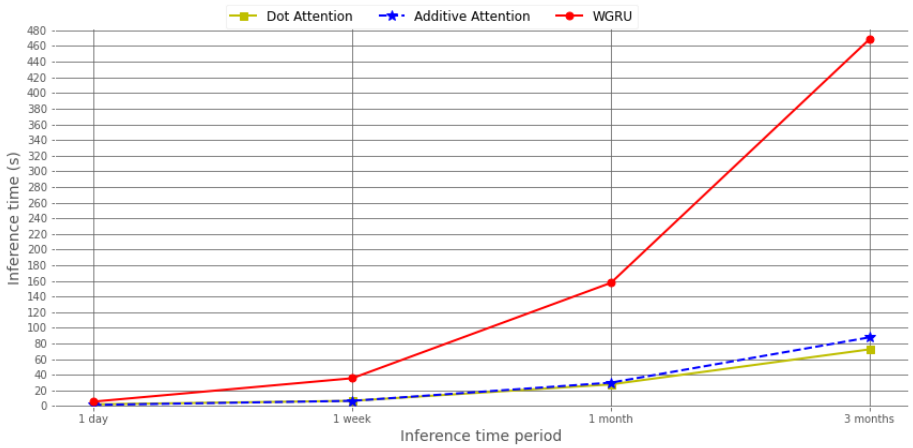

Fig. 3 Inference time versus inference time period for Kettle.

Given 1 day of test data, inference time of WGRU was 5.77 seconds, while SAED-add and SAED-dot achieve 1.56 seconds and 2.27 seconds respectively. As a consequence, SAED-add model is 3.7 times faster than the WGRU and almost 1.5 times faster than the SAED-dot. For 1 week of test data, SAED models are more than 5.2 times faster than the WGRU completing inference in almost 6.8 seconds instead

of 35.6 seconds. Given 1 month of test data, the SAED-dot is 5.7 faster than WGRU with similar test time as the SAED-add. Finally, with test data size of 3 months, the SAED-dot is almost 6.5 times faster than the State of the art and 1.2 times faster than the SAED-add. In this case WGRU inference time was 468.87 seconds versus 72.56 seconds of SAED-dot and 87.89 seconds of SAED-add.

7 Conclusions and Proposals for Future Work

In general, the proposed lightweight SAED models showed good performance and in some cases better results in comparison to the State of the art model (WGRU). Interestingly, the SAED seems to perform better on multi-state devices than on two-state devices. In order to extract more insight on this matter, experiments on different devices should be executed. Furthermore, achieving good performance on the Categories 3 and 4 of experiments, points out the generalization power of the novel architecture. In terms of speed, the SAED models was up to 7.5 and 6.5 faster than the WGRU in training and inference accordingly, resulting that the SAED is more eligible for deployment on embedded systems.

Between the SAED-dot and SAED-add, there is not a clear winner, although the SAED-dot has faster training. Additionally, training and testing on different devices and data sets should be executed in order to evaluate and compare these mechanisms in detail.

It should be mentioned that the architecture of the SAED models could be optimized. SAED was designed after preliminary experiments and tests mainly in order to decide the optimal position of the attention layer. It is suggested that more experiments should be executed considering the number of neurons in CNN, GRU and Dense layers.

To summarize, the use of Attention mechanism granted great generalization ability to a simple and light model, making it possible to achieve good performance in short amount of training and inference times. Therefore, Attention may be used in other architectures in order to improve them in the task of NILM.

References

1. Mahapatra, B.; Nayyar, A. Home energy management system (HEMS): concept, architecture, infrastructure, challenges and energy management schemes. *Energy Syst.* **2019**, doi:10.1007/s12667-019-00364-w.
2. Carrie Armel, K.; Gupta, A.; Shrimali, G.; Albert, A. Is disaggregation the holy grail of energy efficiency? The case of electricity. *Energy Policy* **2013**, *52*, 213–234, doi:10.1016/j.enpol.2012.08.062.
3. Naghibi, B.; Deilami, S. Non-intrusive load monitoring and supplementary techniques for home energy management. In Proceedings of the 2014 Australasian Universities Power Engineering Conference (AUPEC); 2014; pp. 1–5.
4. Parson, O. Hart, G.W., Prototype Nonintrusive Appliance Load Monitor, 1985.

5. Kolter, J.Z.; Jaakkola, T. Approximate Inference in Additive Factorial HMMs with Application to Energy Disaggregation. 11.
6. Kelly, J.; Knottenbelt, W. Neural NILM: Deep Neural Networks Applied to Energy Disaggregation. *Proc. 2nd ACM Int. Conf. Embed. Syst. Energy-Effic. Built Environ. - BuildSys 15* **2015**, 55–64, doi:10.1145/2821650.2821672.
7. Kelly, J.; Knottenbelt, W. The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes. *Sci. Data* **2015**, *2*, 150007, doi:10.1038/sdata.2015.7.
8. Mauch, L.; Yang, B. A new approach for supervised power disaggregation by using a deep recurrent LSTM network. In Proceedings of the 2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP); 2015; pp. 63–67.
9. Kolter, J.Z.; Johnson, M.J. REDD: A Public Data Set for Energy Disaggregation Research. 6.
10. Zhang, C.; Zhong, M.; Wang, Z.; Goddard, N.H.; Sutton, C.A. Sequence-to-point learning with neural networks for nonintrusive load monitoring. In Proceedings of the AAAI; 2018.
11. Krystalakos, O.; Nalmpantis, C.; Vrakas, D. Sliding Window Approach for Online Energy Disaggregation Using Artificial Neural Networks.; 2018; pp. 1–6.
12. Symeonidis, N.; Nalmpantis, C.; Vrakas, D. A Benchmark Framework to Evaluate Energy Disaggregation Solutions. In Proceedings of the Engineering Applications of Neural Networks; Macintyre, J., Iliadis, L., Maglogiannis, I., Jayne, C., Eds.; Springer International Publishing: Cham, 2019; pp. 19–30.
13. Klemenjak, C.; Makonin, S.; Elmenreich, W. Towards Comparability in Non-Intrusive Load Monitoring: On Data and Performance Evaluation. In Proceedings of the 2020 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT); IEEE: Washington, DC, USA, 2020; pp. 1–5.
14. Nalmpantis, C.; Vrakas, D. Machine learning approaches for non-intrusive load monitoring: from qualitative to quantitative comparation. *Artif. Intell. Rev.* **2018**, *52*, doi:10.1007/s10462-018-9613-7.
15. Basu, K.; Debusschere, V.; Seddik, B. Load identification from Power Recordings at Meter Panel in Residential Households.; 2012; pp. 2098–2104.
16. Basu, K.; Debusschere, V.; Seddik, B. Residential Appliance Identification and Future Usage Prediction from Smart Meter.; 2013; pp. 4994–4999.
17. Nalmpantis, C.; Vrakas, D. On time series representations for multi-label NILM. *Neural Comput. Appl.* **2020**, doi:10.1007/s00521-020-04916-5.
18. Nalmpantis, C.; Vrakas, D. Signal2Vec: Time Series Embedding Representation. In; 2019; pp. 80–90 ISBN 978-3-642-54671-6.
19. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27*; Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q., Eds.; Curran Associates, Inc., 2014; pp. 3104–3112.
20. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *ICLR* **2015**.
21. Luong, T.; Pham, H.; Manning, C.D. Effective Approaches to Attention-based Neural Machine Translation. In Proceedings of the Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing; Association for Computational Linguistics: Lisbon, Portugal, 2015; pp. 1412–1421.

22. Cheng, J.; Dong, L.; Lapata, M. Long Short-Term Memory-Networks for Machine Reading. In Proceedings of the Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing; Association for Computational Linguistics: Austin, Texas, 2016; pp. 551–561.
23. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *ICLR* **2015**.
24. Batra, N.; Kelly, J.; Parson, O.; Dutta, H.; Knottenbelt, W.; Rogers, A.; Singh, A.; Srivastava, M. NILMTK: an open source toolkit for non-intrusive load monitoring. In Proceedings of the Proceedings of the 5th international conference on Future energy systems - e-Energy ’14; ACM Press: Cambridge, United Kingdom, 2014; pp. 265–276.