Unlocking the Full Potential of Neural NILM: On Automation, Hyperparameters & Modular Pipelines

Hafsa Bousbiat, Anthony Faustine, Christoph Klemenjak, Lucas Pereira, and Wilfried Elmenreich

***Abstract*—Non-Intrusive Load Monitoring (NILM) techniques are increasingly becoming a key instrument for identifying the power consumption of individual appliances based on a single metering point. Particularly Deep learning models are gaining interest in this regard. However, the challenges brought by NILM datasets and the non-availability of common experimental guidelines tend to compromise comparison, research transparency and replicability. The limited adoption of efficient research instruments and a lack of best practices guidelines contribute in huge part to this problem, where no features, encouraging standardised formats for benchmarking and results sharing, are offered.**

To address these issues, we first present a brief overview of recent best practices for Deep Learning (DL) and highlight how deep NILM research can benefit from these practices. Furthermore, we suggest a novel open-source toolkit leveraging these practices: Deep-NILMTK. The proposed toolkit offers a common testing bed for NILM algorithms independently of the underlying deep learning framework with a modular NILM pipeline that can easily be customised. Furthermore, *Deep-NILMTK* introduces the concept of *Experiment Templating* to offer pre-designed experiments allowing to enhance research efficiency. Leveraging this concept and DL best practices, we present a case-study of creating an online NILM benchmark repository [1] considering eight of the most popular deep NILM algorithms. All sources relative to the tool are made publicly available on Github[2] along with the corresponding documentation.

***Index Terms*—Non-Intrusive Load Monitoring, Load Disaggregation, Deep Neural Networks, NILMTK, Machine Learning, Best Practices**

I. Introduction

NON-INTRUSIVE Load Monitoring (NILM) is a research field investigating approaches producing an estimated breakdown of the total consumption without the need to monitor appliances individually [1]. Deep Neural Networks for NILM (DNN-NILM) gained particular attention from the research community since the first adoption of DL to solve NILM [2] and quickly became the main research stream. Despite the significance of recent contributions, reproducibility and comparability in NILM remain particularly problematic. This is mainly due to the lack of an efficient benchmarking framework with reference algorithms and standardised evaluation procedures [3], [4]. Moreover, the non-availability of source code or its incompatibility with available toolkits makes extended evaluation cumbersome. Consequently, newly proposed NILM approaches are seldomly compared against the same benchmark algorithms [4]. At the same time, scholars investigating new applications of NILM find it hard to quickly adapt and evaluate the most suitable model for the application at hand while considering recent literature.

In addressing the previous problems, several efforts have been made to develop suitable research tools. The *NILMTK* [3] project is the most common open-source software offering a standard interface for algorithm implementation and data analysis. The modular design style of the tool aimed at encouraging scholars to contribute with more benchmark algorithms and enhancing reproducibility of research. It is unarguable that the release on NILMTK-contrib [5] in 2019 represented a turning point in NILM research by introducing deep NILM baselines and a new interface to streamline the addition of new models. Nevertheless, we argue that in the face of the most recent advances in DL research, as it stands, *NILMTK* still falls short in four main aspects:

1) **The NILM pipeline**: Overall, deep learning models for NILM applications share the same pipeline architecture with alteration in the logic of one or several blocks of the pipeline. However, available tools require a new implementation of the whole pipeline for each added model negatively impacting research efficiency.
2) **The non-availability of pre-designed experiments**: A main problem in NILM scholarship is comparability which is related to the different experimental setups adopted by scholars. The current available toolkits do not offer possibilities for scholars to share their experimental design respecting universal standards.
3) **The poor compatibility with popular DL development frameworks:** the high dependency of NILMTK to the deep learning framework used to implement the model prevents comparison with many of recent contributions in the NILM scholarship and prohibits future extension. The first proposals of deep models in NILM were based on TensorFlow and Keras. Nevertheless, researchers in the present are favouring PyTorch to develop their models [6]. While it is not possible to ensure that this trend will continue in the coming years, two significant problems arise from this: 1) the comparability between recent models and existing baselines becomes very challenging, and 2) the new community of scholars,

H. Bousbiat, C. Klemenjak and W. Elmenreich are with the Institute of Networked and Embedded Systems, University of Klagenfurt, Austria.

A. Faustine is with the Department of Information Technology, University of Ghent, Gent, B-9052, Belgium. e-mail: sambaiga@gmail.com

L. Pereira is with ITI, LARSyS, Técnico Lisboa, Portugal. e-mail: lucas.pereira@tecnico.ulisboa.pt

Manuscript received October, 2021;

[1]https://github.com/BHafsa/DNN-NILM-benchmark

[2]https://github.com/BHafsa/deep-nilmtk-v1

[3]https://nilmtk.github.io/

using PyTorch, can not benefit from features offered by *NILMTK*.

4) **The lack of DL best practices:** Advances in the development of methodologies and tools of Machine Learning (ML) [7], [8] resulted in a set of best practices and the emergence of new research fields (e.g., Machine Learning Model Operationalisation Management (MLOps) discipline [9]) for efficient and organised development. Most literature emphasises using best practices such as hyper-parameters optimisation, cross-validation and experiment tracking and management while developing DNN-NILM models. However, their implementation in *NILMTK* requires an extra-coding effort that may be prohibitive due to time constraints.

The current manuscript addresses these issues with three main research contributions:

1) A discussion on existing works towards DL best practices and how these translate to the NILM problem and research in general. More precisely, best practices are provided to what concerns data preparation, experiment management, evaluation processes, and benchmarking procedures.
2) The introduction, for the first time, of deep learning best practices to deep NILM research through the conceptualisation and implementation of a modular toolkit leveraging on the existing efforts to deliver a complete and efficient NILM toolkit (e.g., NILMTK and nilm-contrib). The toolkit is aimed to be fully decoupled from deep learning frameworks, and implements several best practices, including hyper-parameter optimisation, cross-validation, and experiment management.
3) The introduction of the concept of *NILM Experiment Templates* as a means to provide automated and standardised performance evaluation and benchmarks on top of Deep-NILMTK. Furthermore, to show the effectiveness of Deep-NILMTK and of the proposed templating system, a benchmark was developed and made publicly available. The developed benchmark consists of eight different appliances from the UK-DALE dataset that were disaggregated using eight of the available NILM algorithms in the present version of Deep-NILMTK.

The remainder of the paper proceeds as follows: section II discusses recent catalogs of DL best practices and their applicability in NILM research. In section III, we introduce the proposed tool and its different modules. Section IV presents a case study of constructing a NILM benchmark repository leveraging on the features of the proposed toolkit. In section V, we discuss the implications of the proposed toolkit and potential future research opportunities. Finally, section VI concludes with a summary of the main contributions.

II. NILM AND DEEP LEARNING BEST PRACTICES

NILM scholarship is among the youngest fields adopting Deep Neural Networks (DNN), where considerable literature has grown around the topic in the past six years, increasing from two contributions in the year 2015 to more than thirty contributions in 2020 [6]. Despite the thrivingness that DNN-NILM has witnessed, several research gaps remain open. In this respect, NILM scholarship can benefit from the widely acknowledged guidelines of best practices in the DL field to improve the research process, consequently, addressing some of the main challenges present in NILM literature.

The development cycle of DL models is an iterative process composed of several steps, including (1) data preparation, (2) training and evaluation (including hyper-parameters optimisation and model selection), and (3) results reporting [8]. These steps are part of the ML pipeline as defined by recent work [10]. More precisely, these pipelines are composed of a set of sequential blocks allowing to pass from data to ready-to-use models. They help to easily collect, process, transform, and store all the details of the machine learning life cycle [11]. However, despite the fact that the majority of deep NILM models follow either a Sequence-to-Sequence or Sequence-to-Point learning paradigm allowing for easy identification of a common pipeline, the available toolkits paid less attention to this aspect. Consequently, research efficiency is negatively influenced due to extra-coding efforts required which is time and resource consuming, and may lead to unfair comparisons. Moreover, the static implemented NILM pipelines offered in available toolkits constraints scholars on a specific technology and requires high familiarity with the API interfaces.

Furthermore, as a recent review of DNN-NILM [6] highlighted, the impact of different components and parameters in the DNN-NILM pipeline (such as normalisation type, sampling rate, and sequence length) on the overall performance remain unclear. For example, a vital parameter for DNN-NILM is the effective sequence length that was recently proven to influence the disaggregation performance [12] pointing to the fact that it need to be further investigated. Yet, available tools do not enable such studies. In summary, the model development process is often an exploration work that involves selecting a set of hyper-parameters and other experiment settings for the addressed problem. Conducting these studies require complex coding efforts with meticulous attention to the outcome of each step of the experiment.

In addressing the previous problem, it is a best practice, in the development of deep learning models, to automate the hyper-parameters optimisation and model selection [7]. It is also recommended to use experiment tracking and management system [13] for automatic and continuous monitoring and performance measurement [7]. To this end, libraries such as Optuna[4] and Ray-tune[5] for hyper-parameter optimisation and MLFlow[6], Wandb[7], and Neptune[8] are widely used by ML researchers. These tools contribute also in improving the quality of reporting as they provide scholars with detailed description of the experimental setup and the generated predictions. Yet, NILM scholars remain deprived from these tools when using the available NILM toolkits.

[4]https://optuna.org/
[5]https://docs.ray.io/en/master/tune/index.html
[6]https://mlflow.org/
[7]https://wandb.ai
[8]https://neptune.ai

Another major challenge in NILM scholarship is the comparability and replicability of results [6], [14], [15] due to the use of different and non-standardised experimental setups. For example, state-of-the-art performance for each appliance remain unclear due heterogeneous evaluation procedures [6]. The previous issue becomes even harder to address when considering small power appliances investigated only in few contributions. Many NILM toolkits do not integrate any feature that encourages best practices for replicating results, such as automatic management and tracking of experiments. The latter enables a greater degree of automation of experiments, since manual interventions are not required either for the parameter or for the artifact tracking. Second, it allows a complete description of an experimental setup and results relative to its execution in a standardised format that allows easy replication.

When addressing the replicability problem in deep learning scholarship, several best practices catalogs [8], [13] further highlight the importance of releasing source code. Consequently, a platform like *papers with code*[9] has emerged as a one-stop-shop for hosting research papers and their associated code and artifacts. Container-based technology such as Docker and Kubernetes are also becoming popular to prepare and share stable code releases as soon as possible [8]. Moreover, the use of public datasets is highly recommended with continuous sharing of its pre-processed versions, which allows for better traceability in long research projects through data versioning tools (e.g., Data Version Control (DVC)[10]). In the same respect, a second significant recommendation is to supply access to scripts for data cleaning and merging [7], [8]. While the first recommendation encourages the reproducibility, the second aims at enabling the replication of the pre-processing protocols on similar datasets. However, again the adoption of these tools in NILM remains very limited. The recently introduced NILMTK-API [5] was a massive step in this regard. However, the toolkit does not include any features for sharing pre-designed experiments in a universal repository which does not encourage the sharing culture.

III. DEEP-NILMTK

Deep-NILMTK is an open-source toolkit for deep NILM models gathering vital tools from both worlds: NILM research and the DL field. It implements a generic modular NILM pipeline totally decoupled from the deep learning technologies. Thus, it extends existing frameworks and allows for easy future extensions.

Deep-NILMTK is designed around three main goals. First, it aims at offering a more inclusive toolkit that benefits scholars independently from the deep learning framework they are using. Second, it allows a straightforward customisation. Third, it helps scholars manage and speed up their work by offering experiment templates and implementing DL best practices.

To this end, the present version of *Deep-NILMTK* is compatible with the two most popular deep learning frameworks (PyTorch and TensorFlow), while offering the possibility of integrating other frameworks. Considering the PyTorch framework, the toolkit incorporates 8 deep NILM baselines; 5 ported from NILMTK-contrib versions and 3 more recent deep models that were specifically implemented for this tool. Considering the Tensorflow framework, the toolkit contains two baselines where we plan to extend it in future work with more recent models as suggested in [16]. Furthermore, *Deep-NILMTK* incorporates three best practices for DL discipline that we argue are urgently needed in DNN-NILM. More precisely: (1) time-series cross-validation, (2) hyper-parameter optimisation, and (3) automatic experiment tracking. In summary, the tool allows NILM scholars to benefit from recent contributions in NILMTK [17] and its high-level API [5] while offering the possibility of using current best practices in the DL discipline to develop, evaluate and manage experimental studies efficiently and independently of deep learning technologies.

A. The NILM Pipeline in Deep-NILMTK

Deep-NILMTK implements a modular pipeline inspired by modern deep learning pipelines allowing to overcome the burdens of traditional software engineering practices. The implemented pipeline is illustrated in Fig. 1. It was designed as a sequence of loosely coupled blocks giving freedom to scholars to investigate different setups with minimal coding effort. Each block in the pipeline has its own interface and can be altered with a custom one respecting the same interface without impacting the whole pipeline. This design choice has many advantages. First, it allows for fast prototyping where only the parts of interest from the pipeline must be re-implemented. Second, using a universal pipeline allows for fair comparison and benchmarking in NILM scholarship that can come as a first step towards solving the comparability problem highlighted several times in related work [3], [18]. Finally, it facilitates and encourages replicability where the models can be easily tested on extended datasets to gain further insights.

The first block of the implemented pipeline is a pre-processing block in which only the aggregate power is normalised. It is followed by a feature generation block that allows, for example, to derive temporal features from the aggregate power series. The generated features serve as an input to a high-level trainer that enables the management of experiments independently of the deep learning framework. This high-level trainer is endowed with a low-level trainer implementing a specific interface that decouples the whole NILM pipeline from the deep learning technology used to implement the model. One low-level trainer is required for each new deep learning framework to be compatible with Deep-NILMTK and allows to define: (1) framework-dependent data loaders that are expected to normalise the target power consumption, (2) initialise the network models, (3) perform training for one model, and finally generate predictions. We note that leaving the target normalisation to the data loader is justified with the fact that some models require the derivation of ON/OFF states from the power values that can be only done with the original power series (e.g., BERT4NILM [19]).

[9]https://paperswithcode.com/

[10]DVC is built to make ML models shareable and reproducible. It is designed to handle large files, data sets, ML models, and metrics as well as code:https://dvc.org/

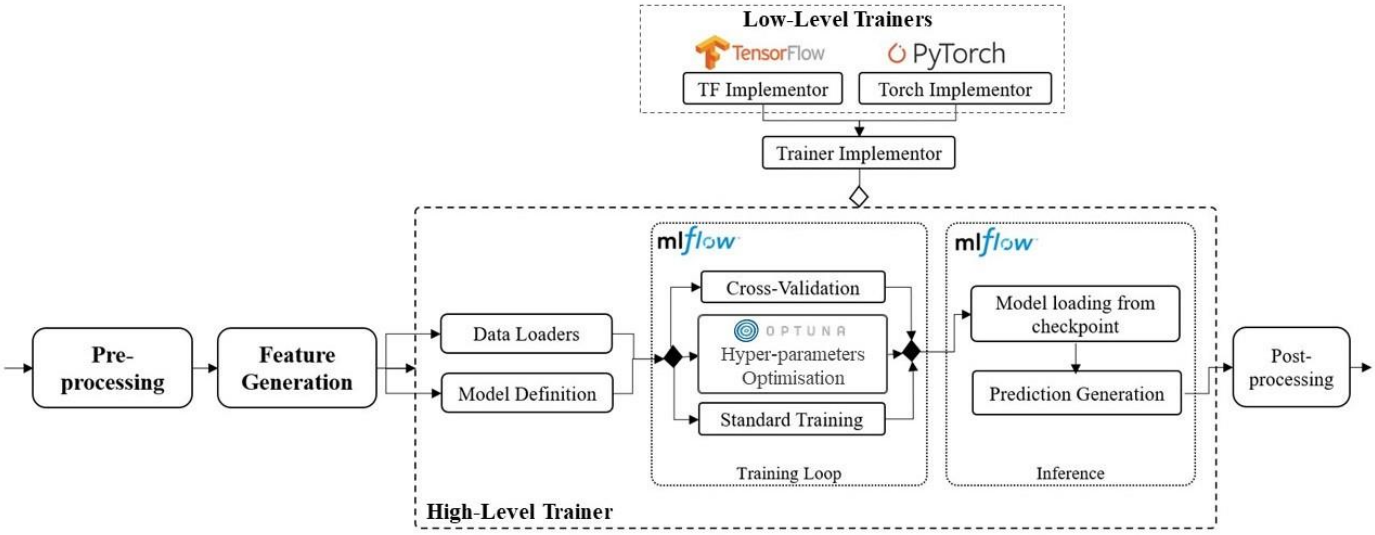


Fig. 1. Overview of the NILM pipeline implemented in *Deep-NILMTK*

The high-level trainer uses the interface as a tool to interact with deep learning frameworks and their APIs. It is composed of two sub-components. The first component is responsible for training the models. It allows for three training strategies: (1) a standard training strategy [3], [20] which consists of training the models by splitting the training data into an 80% for training and 20% for validation, (2) a training strategy using k-fold time-split cross-validation [21], and finally (3) a hyperparameter optimisation strategy leveraging on Optuna [22]. The second component is responsible for generating predictions using the pre-trained models. The different steps of both components of the high-level trainer are tracked and logged using the automatic logging interface of MLFlow. The last block in the pipeline is the post-processing block responsible for restoring the power values from the generated predictions and performing sequence aggregation in the case of sequence-to-sequence models.

B. Experiment Templating

The evaluation of deep learning models relies mainly on the availability of benchmarks to assess the contributions against existing literature. This aspect remains challenging even in mature fields. In addition to this challenge, it remains hard to directly compare models trained and tested on the different buildings or even different periods from the same dataset. Moreover, scholars tend to consider only parts from the available data to evaluate new contributions due to limited resources. Therefore, the characteristics of data selection are an essential aspect of comparability and benchmarking, where it becomes mandatory to retrain the baseline models each time.

To address these challenges, Deep-NILMTK introduces the concept of *Experiment Templating* in NILM. It allows scholars to have the same testbed and to design and share their templates. Experiments templating is built upon the recently introduced NILMTK API [5]. It allows to pre-define the parameters of the API except for the algorithms. Such pre-configured experiments can be used to benchmark existing baselines and evaluate new contributions. Hence, these templates allow fair comparison and encourage the culture of sharing and collaboration while saving both time and computational resources.

C. Deep Learning Best Practices in Deep-NILMTK

The deep learning discipline thrived in the last decade, where a variety of tools were suggested to assist the development process, among which many are relevant for research setups in general and the NILM scholarship in particular. Deep-NILMTK incorporates three of the most commonly used and needed tools for training deep learning models, detailed in the following:

1) ***Cross-Validation****:* Splitting the data into separate train, validation, and testing sets is typical to train deep NILM models where each set consists of non-overlapping contiguous data. Nonetheless, this training approach remains limited in addressing some evaluation criteria (e.g., generalisability) compared to time series cross-validation providing more robust and representative results. *Deep-NILMTK* implements k-folds time-split cross-validation[11] where the training data is split into k sets, and the model is trained k times in which the successive training sets are supersets of those that come before them. The performance results are obtained considering all the models to provide a clearer picture of the performance variation over different sets.

2) ***Hyper-parameters Optimisation****:* Hyper-parameter optimisation is important for the development of deep learning models requiring expertise and extensive trial and error [22]. *Deep-NILMTK* implements automatic hyper-parameter optimisation to address this challenge, leveraging the Optuna framework [22]. The implemented hyper-parameter search provides a mechanism where unpromising trials are stopped

[11]https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.TimeSeriesSplit.html

at the early stages leading to a faster exploration of the considered space.

3) ***Experiment Tracking and Management****:* Traditional experiment tracking using spreadsheets and text files to keep track of performed studies tends to be error-prone and quickly becomes challenging to handle [23]. To overcome these issues several solutions were introduced in the ML discipline (e.g., *Wandb*, *Neptune*) among which only two are open-source *Tensorboard*, and *MLflow*. *Tensorboard* is more suitable for monitoring a single experiment as its interface becomes challenging to interpret with increasing number of experiments, leading thus to the selection of *MLflow*.

MLflow contains several APIs, including a tracking API that allows easy management of experiments. *Deep-NILMTK* makes extensive use of this API to trace training metrics. The pieces of code calling this API were inserted such that all the experiments are organised according to appliances, with each experiment gathering multiple runs related to different models. The organisation above was preferred to enable the comparison of the disaggregation performance brought by different models on a per-appliance basis. The *MLflow* user interface allows visualising a detailed summary of each experiment in a very user-friendly interface.

IV. CASE STUDY: CREATING A NILM BENCHMARK

In this section, we present a case study to demonstrate the strength and effectiveness of adopting deep learning best practices in developing NILM research, and at establishing a benchmark that can be used by other researchers to assess their algorithms without extra re-training requirements.

To state more precisely, the benchmark comprises: 1) a predefined set of experiments (i.e., experiment templates) that scholars can directly use and (2) a shared repository containing the detailed description of the performed experiments. At first, the templates are used to evaluate the disaggregation performance of the considered algorithms with an input window length of $1208s$ (20 minutes). However, to gain further insights about the effect of the input size length, we investigate this parameter in the case of three different appliances, two appliances with high power values and one appliance with small power values. The range of the window size was set to vary between $792s$ (13 minutes) and $4480s$(74 minutes). Only one baseline model, the *sequence-to-point*, is considered during this last experiment.

The main goals of the presented case-study are as follows:

1) Offering re-usable experimental designs for different appliances, considering the UKDALE dataset.
2) Evaluating DNN-NILM baselines and recent models in the case of appliances with heterogeneous power consumption magnitudes.
3) Offering an open-source DNN-NILM benchmark for the considered appliances.

A. Definition of the Experiment's Template

To design a re-usable set of experiments that scholars can directly adopt during evaluation, the concept of experiment-templates previously introduced was adopted. We suggest a set of templates for the different appliances that are considered. All these templates are based on data from the UK-DALE [24] dataset. UK-DALE contains a record of domestic energy consumption from five houses in the UK, including the aggregated and sub-metered consumption for individual appliances. For each appliance an analysis of the data from building 1 of UKDALE was performed and the longest good recorded section was used as training data. Appliances with similar periods were grouped in the same template. For all appliances, we consider active power data from the first building of the UKDALE dataset, and a common testing period of one month (from 2015-04-16 to 2015-05-15) at a sampling rate of $8s$. All templates consider real aggregate power instead of synthetic aggregate power where gaps in the data are dropped. Table I illustrates the suggested templates.

TABLE I
NILM TEMPLATES FOR UKDALE DATASET

ID	Training data		appliances	threshold
	start	end		(*watts*)
0	07/09/2013	13/11/2013	coffee maker (CM)	10
1	13/03/2014	21/07/2014	washing machine (WM)	1000
			microwave (MW	600
			dryer (DRY)	1000
2	16/08/2014	04/10/2014	kettle (KTL)	1000
3	26/04/2016	30/07/2016	dish washer (DW)	1000
			television (TV)	50
			audio amplifier (AA)	10

Two metrics were used to evaluate the candidate algorithms, the Normalised Disaggregation error (NDE) and the F1-score, defined as per the following equations:

$$NDE = \frac{\sum (y_t - \hat{y_t})^2}{\sum \hat{y_t}^2} \tag{1}$$

$$F1 - score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{2}$$

Where the Precision = TP/(TP+FP), Recall = TP/(TP+FN). Moreover, the thresholds used to define the confusion matrix for each appliance are illustrated in the last column of table I. The NDE metric was favored as it offers a normalised measure enabling fair comparison between appliances with different power magnitudes.

B. The Benchmarked Deep NILM Algorithms

In the scope of the presented benchmarking study, we only consider models implemented in PyTorch where five DL baselines, that were originally developed for NILMTK, are available with three recent contributions, namely UNET-NILM, BERT4NILM [19], SAED [25]. Table II provides more details about each of the benchmarked algorithms. We leverage on an adaptive learning rate of 10^{-4} and using an Adam optimiser. The code of the three recent models was directly adopted from the original repositories in Github and integrated in the toolkit.

TABLE II
OVERVIEW OF THE PYTORCH BENCHMARK ALGORITHMS

	Output's format		Type of layers	State est.
	S2S	S2P		
Seq2Seq	✓		Conv1D	
Seq2Point		✓	Conv1D	
DAE	✓		Conv1D	
RNN		✓	LSTM	
GRU		✓	GRU	
SAED [25]		✓	attention	
BERT4NILM [19]	✓		attention	✓
UNET-NILM [26]		✓	Conv1D	✓

C. Results

The disaggregation results obtained for different appliances are illustrated in Table III for the two considered metrics. The first five appliances are considered among the big consuming appliances ($\geq 1000watts$) and have been extensively investigated in the literature. The last four appliances are considered low power consuming appliances ($\leq 300watts$). On the other hand, the models are organised according to the output's shape.

Considering the first five appliances, it is remarkable to observe that the majority of models provide very competitive results. The reported values of the NDE demonstrate that Seq2Point models provide best results in the case of the WM, the DRY and the DW, with the superiority of the GRU baseline in the latter case. On the other hand, they reveal that Seq2Seq models provide better results in the case of the KTL and the MW. A careful observation of the obtained results illustrates that UNET model provide a good compromise between both options since it provides approximate results to the best models for all the five appliances. On the other hand, the two DNN-NILM algorithms relying on the attention mechanism demonstrated acceptable results. Yet, they failed to outperform the existing baselines. These results are further confirmed with the values of the F1-score where all reported values were higher than 70%, except in the case of DAE for some appliances and the MW with some models.

Interestingly, the DAE baseline failed to provide acceptable results for all appliances during the testing phase for both metrics. One justification for this results could be related to the input window length used, specially in the case of the MW and the KTL that operate only for few minutes. On the other hand, the results obtained in the case of the MW could be related to the training data. The experimental template including the MW also includes two other big consuming appliances, the WM and the DRY, that may have shadowed the MW activations.

In summary, the obtained results in this first step demonstrate the good quality of the training data selected for all appliances, except the MW, since it allowed the majority of the models to provide acceptable results. It also illustrates that the recently proposed models based on the attention mechanism still need to be further investigated and developed to take full advantage of the benefits of this type of layer. Finally, the results suggest that the UNET model can provide very good results in the case of big consuming appliances with a maximum NDE of 0.40 and a minimum f1-score of 90%, disregarding the MW.

The obtained results for the remaining appliances reveal that deep NILM models still need to be further adapted for the case of small consuming appliances. The obtained results for these appliances are merely acceptable. Fig. 2 presents an example of the generated activation by the seq2point baseline for the TV and the KTL. As the figure illustrates, the models succeeded in both cases to detect an activation with some errors in estimating the exact power consumption. However, while an error in the range of 60-100 Watt is negligible in the case of the KTL, it remains significant in the case of the TV as it could translate into an ON or an OFF event.

To further evaluate DNN-NILM models in the case of appliances with low power values, the effect of the sequence length on the overall performance is studied considering the Seq2Point baseline. Fig. 3 illustrates the best validation loss obtained for different input window sizes in the case of three appliances: the KTL, the MW and the TV. As the figure demonstrates, the model succeeds to achieve smaller validation losses during training with small window sizes in the case of the KTL and the MW ($\leq 2000s$). Nonetheless, bigger window sizes provide smaller validation losses in the case of the TV. It is thus to conclude that the results provided in Table III in the case of the last three appliances remain limited and further ablation studies are required to obtain a fair comparison.

All the obtained models and details of the conducted experiments are made publicly available as an attempt to establish a universal NILM benchmark repository. The latter may serve as reference for future work allowing to avoid re-training efforts.

V. DISCUSSION

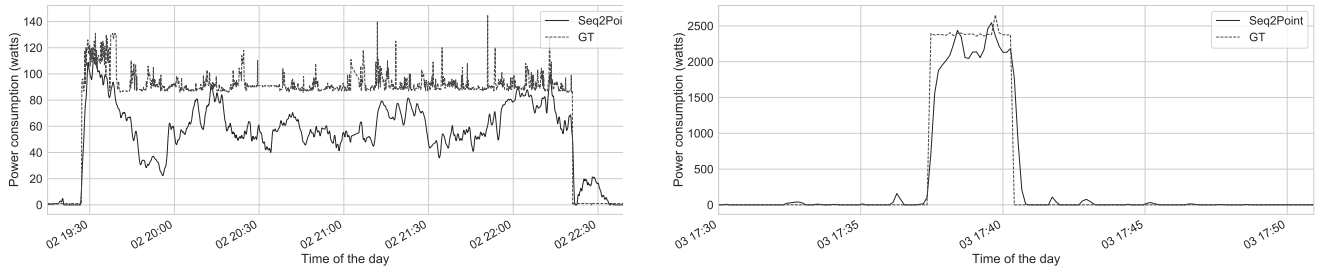
The adoption of deep learning best practices in NILM scholarship has the potential to provide the answer to many of the problems that have long been present in the NILM scholarship such as reproducibilty and comparability. The proposed toolkit leverages on these guidelines to support and standardise NILM research experiments.

Deep-NILMTK is the first toolkit for deep learning models focused on the NILM pipeline itself. It is inspired by tools from other fields (e.g., [27]) and is aimed as a remedy to many problems in the deep NILM scholarship, mainly reproducibility and benchmarking that enable a fast and transparent research process. We argue that Deep-NILMTK is a first step towards gathering the NILM community working with different deep learning technologies under the same umbrella for a more inclusive, transparent and reproducible research. Moreover, the modular pipeline offered in the toolkit allows for fast prototyping and easy extension with minimal coding efforts which promotes a time-efficient research process. It thus offers a unified testing bed to evaluate new NILM algorithms promoting fair comparisons and easy bechmarking.

On the other hand, the introduced *Experiment Templating Concept* combined with MLOps tools implemented in Deep-NILMTK offers a tool to overcome the non-availability of common experimental guidelines that often compromises comparison [3]. Furthermore, it helps to avoid redundant experimenting as scholars have the opportunity to share the details of their experiments with the community using a standardised

TABLE III
DISAGGREGATION RESULTS FOR THE DIFFERENT APPLIANCES

	NDE								f1-score							
	S2S			S2P					S2S			S2P				
	S2S	DAE	BERT	S2P	RNN	GRU	UNET	SAED	S2S	DAE	BERT	S2P	RNN	GRU	UNET	SAED
WM	.30	.58	.48	**.26**	.28	.33	.28	.43	**.97**	.79	.83	**.96**	.95	.93	.95	.85
DW	.43	.62	.77	.59	.58	**.37**	.40	.71	.91	.70	.85	.78	.79	**.92**	.90	.73
DRY	.30	.58	.40	**.26**	.31	.39	**.27**	.47	**.97**	.80	.89	**.96**	.94	.90	**.95**	.81
KT	**.36**	.92	.48	.44	.58	.40	**.37**	.46	.93	.0	**.91**	.88	.80	.89	**.91**	.86
MW	**.58**	.89	.88	.62	.76	.75	.67	.80	**.77**	.12	.28	**.75**	.61	.60	**.72**	.52
TV	.63	.66	.96	.57	.73	.65	.57	.64	.86	.72	.0	.80	.72	.73	.51	.74
AA	.77	.81	1.0	.81	.83	.78	.84	.79	.56	.50	.0	.52	.52	.54	.50	.54
CM	.98	1.0	0.00	.94	1.0	1.0	.97	2.5	.56	.56	.0	.56	.56	.56	.56	.55



(a) Television (b) The kettle

Fig. 2. One generated activation by the Seq2Point baseline

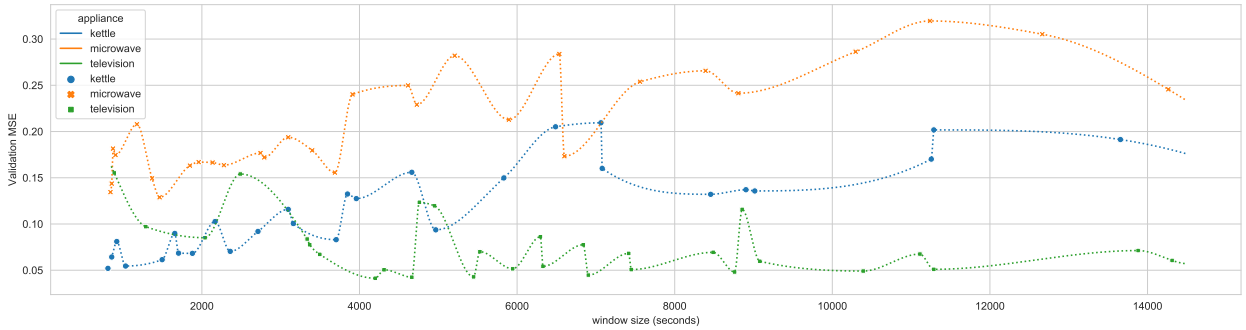


Fig. 3. The results of window size optimisation for the kettle, the microwave and the television using Sequence-to-Point baseline.

format for a more transparent and collaborative research. This would not only enhance the comparability problem in the NILM scholarship but also enable research groups with limited resources to put more focus on their contribution and overcome the requirement of massive computing clusters and highly specialised hardware.

Furthermore, we argue that Deep-NILMTK could be the first step towards building a Neural Architecture Search (NAS) benchmark dataset containing network performance across different templates. While such datasets are very popular in other disciplines [28], they received less attention from the NILM community due to the non-availability of tools encouraging such practices. To the best of our knowledge, the current manuscript is the first to offer a toolkit that allows easy construction of such datasets as it was demonstrated in the presented case-study. These datasets allow for easy literature reviews and fast identification of state-of-the-art models in the NILM scholarship. Future work may take it even further and offer online platforms gathering the results of different researchers using Deep-NILMTK, as we believe that having the results visible and auditable, would improve transparency. These platforms can allow also industrial partners and energy retailers to benefit from the most recent results in the NILM research.

Even though *Deep-NILMTK* could be the springboard for using best practices of DL in the NILM scholarship and the first to suggest a universal toolkit for deep learning NILM, it has some limitations. First, Deep-NILMTK is focused only on eventless algorithms. Future versions of the tool should also offer a unified interface for event-based models relevant

in industrial settings (e.g., [29]). Second, the presented case study remains limited and can only be considered as a proof of concept for the implemented toolkit as well as the potential brought by the offered features. Future studies extending the provided bechmarking repository are therefore mandatory to achieve the main goals of the toolkit.

VI. Conclusion

The work at hand introduced a new NILM toolkit building on existing toolkits and oriented towards enhancing the research progress through machine learning inspired best practices. In its current version the toolkit implements a modular NILM pipeline that is fully independent from deep learning frameworks, which allows for the first time to offer a universal testing bed for the NILM algorithms. We also offer an online benchmarking repository along with a predesigned experiment setup that promotes fair comparison and efficiency in research. To the best of our knowledge, the current contribution is the first to bring concrete solutions enabling scientists in the NILM scholarship to address the comparability issues and overcome the non-availability of common experimental guidelines.

The current contribution is a first step in a larger initiative that is required in the NILM scholarship to encourage transparent and efficient research. Further efforts leveraging on the concepts of meta-research, concerned about exploring techniques to improve the research it self [30], could hold the keys to achieve major advances in developing transparent and robust NILM algorithms.

Acknowledgments

This work was supported by the DECIDE doctoral college, Digital Age Research Center (D!ARC).

Lucas Pereira has received funding from the Portuguese Foundation for Science and Technology (FCT) under grants CEECIND/01179/2017 and UIDB/50009/2020.

References

[1] G. Hart, "Non-Intrusive Appliance Load Monitoring." *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.

[2] J. Kelly and W. Knottenbelt, "Neural nilm: Deep neural networks applied to energy disaggregation," in *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*. ACM, 2015, pp. 55–64.

[3] C. Klemenjak, S. Makonin, and W. Elmenreich, "Towards comparability in non-intrusive load monitoring: On data and performance evaluation," in *2020 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. IEEE, 2020.

[4] A. Faustine, N. H. Mvungi, S. Kaijage, and K. Michael, "A survey on non-intrusive load monitoring methodies and techniques for energy disaggregation problem," *CoRR*, vol. abs/1703.00785, 2017. [Online]. Available: http://arxiv.org/abs/1703.00785

[5] N. Batra, R. Kukunuri, A. Pandey, R. Malakar, R. Kumar, O. Krystalakos, M. Zhong, P. Meira, and O. Parson, "Towards reproducible state-of-the-art energy disaggregation," in *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, 2019, pp. 193–202.

[6] P. Huber, A. Calatroni, A. Rumsch, and A. Paice, "Review on deep neural networks applied to low-frequency nilm," *Energies*, vol. 14, no. 9, p. 2390, 2021.

[7] A. Serban, K. van der Blom, H. Hoos, and J. Visser, "Adoption and effects of software engineering best practices in machine learning," in *Proceedings of the 14th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2020, pp. 1–12.

[8] M. Lindauer and F. Hutter, "Best practices for scientific research on neural architecture search," *Journal of Machine Learning Research*, vol. 21, no. 243, pp. 1–18, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-056.html

[9] S. Mäkinen, H. Skogström, E. Laaksonen, and T. Mikkonen, "Who needs mlops: What data scientists seek to accomplish and how can mlops help?" *CoRR*, vol. abs/2103.08942, 2021. [Online]. Available: https://arxiv.org/abs/2103.08942

[10] S. Amershi, A. Begel, C. Bird, R. DeLine, H. Gall, E. Kamar, N. Nagappan, B. Nushi, and T. Zimmermann, "Software engineering for machine learning: A case study," in *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. IEEE, 2019, pp. 291–300.

[11] M. Aiswarya Raj, J. Bosch, H. H. Olsson, and A. Jansson, "On the impact of ml use cases on industrial data pipelines," in *2021 28th Asia-Pacific Software Engineering Conference (APSEC)*, 2021, pp. 463–472.

[12] A. Reinhardt and M. Bouchur, "On the impact of the sequence length on sequence-to-sequence and sequence-to-point learning for nilm," in *Proceedings of the 5th International Workshop on Non-Intrusive Load Monitoring*, 2020, pp. 75–78.

[13] V. A. Makarov, T. Stouch, B. Allgood, C. D. Willis, and N. Lynch, "Best practices for artificial intelligence in life sciences research," *Drug Discovery Today*, 2021.

[14] S. S. Hosseini, K. Agbossou, S. Kelouwani, and A. Cardenas, "Non-intrusive load monitoring through home energy management systems: A comprehensive review," *Renewable and Sustainable Energy Reviews*, vol. 79, pp. 1266–1274, 2017.

[15] A. Ruano, A. Hernandez, J. Ureña, M. Ruano, and J. Garcia, "Nilm techniques for intelligent home energy management and ambient assisted living: A review," *Energies*, vol. 12, no. 11, p. 2203, 2019.

[16] H. Shastri and N. Batra, "Neural network approaches and dataset parser for nilm toolkit," in *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, 2021, pp. 172–175.

[17] N. Batra, J. Kelly, O. Parson, H. Dutta, W. Knottenbelt, A. Rogers, A. Singh, and M. Srivastava, "Nilmtk: An open source toolkit for non-intrusive load monitoring," in *Proceedings of the 5th international conference on Future energy systems*, 2014, pp. 265–276.

[18] L. Pereira and N. Nunes, "Performance evaluation in non-intrusive load monitoring: Datasets, metrics, and tools—a review," *Wiley Interdisciplinary Reviews: data mining and knowledge discovery*, vol. 8, no. 6, p. e1265, 2018.

[19] Z. Yue, C. R. Witzig, D. Jorde, and H.-A. Jacobsen, "Bert4nilm: A bidirectional transformer model for non-intrusive load monitoring," in *Proceedings of the 5th International Workshop on Non-Intrusive Load Monitoring*, 2020, pp. 89–93.

[20] C. Klemenjak, S. Makonin, and W. Elmenreich, "Investigating the performance gap between testing on real and denoised aggregates in non-intrusive load monitoring," *Energy Informatics*, vol. 4, no. 1, pp. 1–15, 2021.

[21] C. Bergmeir and J. M. Benítez, "On the use of cross-validation for time series predictor evaluation," *Information Sciences*, vol. 191, pp. 192–213, 2012.

[22] A. Takuya, S. Shotaro, Y. Toshihiko, O. Takeru, and K. Masanori, "Optuna: A next-generation hyperparameter optimization framework," in *Proc. 25rd ACM SIGKDD Int. Conference on Knowledge Discovery and Data Mining*, 2019.

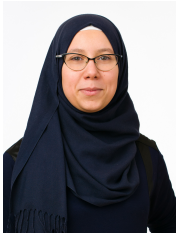
[23] M. M. John, H. H. Olsson, and J. Bosch, "Towards mlops: A framework and maturity model," in *2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 2021, pp. 1–8.

[24] J. Kelly and W. Knottenbelt, "The uk-dale dataset, domestic appliance-level electricity demand and whole-house demand from five uk homes," *Scientific data*, vol. 2, no. 1, pp. 1–14, 2015.

[25] N. Virtsionis-Gkalinikis, C. Nalmpantis, and D. Vrakas, "Saed: Self-attentive energy disaggregation," *Machine Learning*, pp. 1–20, 2021.

[26] A. Faustine, L. Pereira, H. Bousbiat, and S. Kulkarni, "UNet-NILM: A Deep Neural Network for Multi-tasks Appliances State Detection and Power Estimation in NILM," in *Proceedings of the 5th International Workshop on Non-Intrusive Load Monitoring*, ser. NILM'20. New York, NY, USA: Association for Computing Machinery, Nov. 2020, pp. 84–88.

[27] Y. Mehta, C. White, A. Zela, A. Krishnakumar, G. Zabergja, S. Moradian, M. Safari, K. Yu, and F. Hutter, "Nas-bench-suite: Nas evaluation is (now) surprisingly easy," in *International Conference on Learning Representations*, 2021.
[28] C. Vieira, L. P. Cáceres, and L. C. T. Bezerra, "Evaluating anytime performance on nas-bench-101," in *2021 IEEE Congress on Evolutionary Computation (CEC)*, 2021, pp. 1249–1256.
[29] C. Athanasiadis, D. Doukas, T. Papadopoulos, and A. Chrysopoulos, "A Scalable Real-Time Non-Intrusive Load Monitoring System for the Estimation of Household Appliance Power Consumption," *Energies*, vol. 14, no. 3, p. 767, Jan. 2021.
[30] J. P. Ioannidis, "Meta-research: Why research on research matters," *PLoS biology*, vol. 16, no. 3, p. e2005468, 2018.

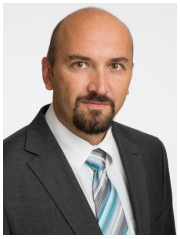


Hafsa Bousbiat was born in Algiers, Algeria. She received her Engineering and Master degree in computer science from the *Ecole Nationale Supérieure d'Informatique (ESI ex.INI)*, Algiers. She is currently pursuing a PhD in Information and Communications Engineering at the University of Klagenfurt, Austria. Since 2019, she is a senior researcher within the Digital Age Research Center (D!ARC). The main focus of her work is the evaluation of the ability of ICT systems based on NILM to influence decision making in smart homes through improving the performance of load disaggregation and understanding user's concerns and requirements for such services.

Anthony Faustine is a machine learning researcher and practitioner with experience applying data analytics and machine learning techniques to business problems. He received a B.sc. Degree in Electronics Science and Communication from the University of Dar es Salaam, Tanzania, and the M.sc. Degree in Telecommunications Engineering from the University of Dodoma in 2010. He is currently pursuing a Ph.D. Degree in Computer Science Engineering at Further Energy and Environment Research Laboratory (FEELab), ITI/LARSyS, Tecnico Lisboa in Portugal. His research interest is in robust machine learning algorithms for future energy systems focusing on load forecasting, NILM and data-driven energy optimisation. Anthony also works as a Lead data scientist at the Centre for Intelligent Power at Eaton's global headquarters in Dublin, where he develops data science-based product features to address emergent use cases in the industrial and energy domains.

Christoph Klemenjak was born in Villach, Austria. He received his Masters degree in Information and Communications Engineering from the University of Klagenfurt, where he currently pursues a Ph.D. Before focussing on industrial applications of Applied Machine Learning, he worked as research and teaching assistant at the Institute of Networked and Embedded Systems, part of the University of Klagenfurt. His research activities focus on Applied Machine Learning in Smart Microgrids, Load Disaggregation and Embedded Machine Learning for Smart Metering. He published papers at several top tier ACM and IEEE conferences as well as journals focusing on Energy Informatics.

Lucas Pereira received his Ph.D. in Computer Science from the University of Madeira, Portugal, in 2016. Since then, he is at ITI/LARSyS, where he leads the Further Energy and Environment research Laboratory (FEELab). Since 2019 he is a research fellow at Técnico Lisboa. Lucas's research applies data science, machine learning, and human-computer interaction techniques towards bridging the gap between laboratory and real-world applicability of ICT for the sustainable development goals (SDGs). His current research focuses on future energy systems and sustainable built environments and it typically involves the real-world deployment and evaluation of monitoring technologies and software systems.



Wilfried Elmenreich received the Dr.Techn. degree from the Vienna University of Technology, Vienna, Austria, in 2002. He is a University Professor of Smart Grids with the Institute of Networked and Embedded Systems, Alpen-Adria-Universität Klagenfurt, Klagenfurt, Austria, where he moved to the Alpen-AdriaUniversität Klagenfurt as a Senior Researcher in 2007. After, a visiting professorship with the University of Passau, Passau, Germany, in 2013, he followed the call to the University of Klagenfurt. He is a member of the Senate at the Alpen-Adria-Universität Klagenfurt, a Counselor of the IEEE Student Branch, and is involved in the master program on Game Studies and Engineering. He has authored of several books and has published over 200 articles in the field of networked and embedded systems. His research interests include intelligent energy systems, self-organizing systems, and technical applications of swarm intelligence.