Neural network approaches and dataset parser for NILM toolkit

Hetvi Shastri
hetvi.shastri@iitgn.ac.in
IIT Gandhinagar
India

Nipun Batra
nipun.batra@iitgn.ac.in
IIT Gandhinagar
India

ABSTRACT

Non-intrusive load monitoring (NILM) involves separating the household aggregate energy consumption into constituent appliances. In 2014, a toolkit called NILMTK was released towards making NILM reproducible. Subsequently, in 2019, an improved version called NILMTK-contrib, focused on experiments and ease of adding new algorithms was released. Since then, there have been significant advances in neural networks for various applications, and in the NILM domain. In this paper, we implement five recent neural network architectures for NILM in NILMTK-contrib and benchmark against existing algorithms. Further, in this paper, we also implement a dataset parser for a publicly available dataset called IDEAL containing 255 homes with 39 homes having appliance data. We find that the new algorithms are comparable or better than the state-of-the-art over a subset of the appliances.

CCS CONCEPTS

• **Computing methodologies → Machine learning algorithms**.

KEYWORDS

datasets, neural networks, NILM, energy disaggregation

ACM Reference Format:
Hetvi Shastri and Nipun Batra. 2021. Neural network approaches and dataset parser for NILM toolkit. In *The 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '21), November 17–18, 2021, Coimbra, Portugal.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3486611.3486652

1 INTRODUCTION

Non-intrusive load monitoring (NILM) is the task of estimating power consumption of appliances from aggregate *mains* level reading [7]. The problem was initially presented in the 1980s. The field of NILM has seen a recent renewed interest due to the availability of smart meter data, improved compute and algorithmic capabilities.

Due to different sampling periods of datasets, different metrics used for different algorithms, and the non-availability of public benchmarks, there was a problem in comparing various algorithms and datasets. In 2014, NILMTK [2], an open-source toolkit, helped the community to make comparative analyses among various algorithms and datasets. In 2019, an improvement for NILMTK called NILMTK-contrib [5] proposed a new easier experiment API and state-of-the-art neural network baselines.

Since 2019, a number of neural networks based NILM approaches have been proposed. Our first contribution in this paper is an implementation of five such methods. These algorithms leverage the recent advances in attention [1, 15], residual networks [8, 9], and multi-task learning [6, 15]. The intuition behind attention weights is to focus on relevant input information. The addition of residual network proposed that the presence of the skip connection as shown in Figure 1 allows "deeper" learning. The model with regression and classification subnetwork as shown in Figure 2 is simultaneously trained on both the subnetworks is inspired by advances in multi-task learning. The classification sub-network provided a sub-path for detecting the on-off state of an appliance accurately. Our second contribution in this paper consists of a parser for the publicly available dataset from the UK called IDEAL [16].

Our third contribution in this paper is the empirical comparison of five proposed neural network approaches against the existing state-of-the-art implementations on two publicly available datasets: REDD [13] and IDEAL [16]. Our main findings are: i) For a subset of appliances, the newly implemented algorithms performed better than the existing ones; ii) Existing algorithms mostly always predicted the sparse appliances as off, while newly implemented algorithms performed better capturing the on-off state.

2 IMPLEMENTATION OF NEURAL NETWORK METHODS FOR NILM

We have implemented five different types of neural networks and contributed them to the NILMTK-contrib repository. This section contains model overview for various algorithms and explains the importance of different layers in corresponding networks. These algorithms are inspired broadly by the Seq2Seq and Seq2Point [18] architectures which accept a sequence of mains reading as input and predict a sequence or point for the appliance reading.

2.1 Bidirectional Long short term memory (BiLSTM)

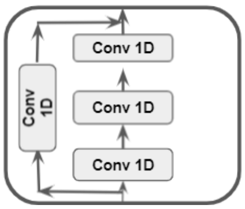
Recurrent neural networks can learn patterns from sequential data. However, they are limited to relatively short-term context. Long short term memory model (LSTM) and Bidirectional LSTM (BiLSTM) are used for learning relatively longer-term context in one and two directions respectively. It should be noted that while a relatively simple model, we introduce it here to set the context for improvements over LSTMs presented below. As proposed in prior work [11], the architecture of the model is as follows :

BiLSTM Model

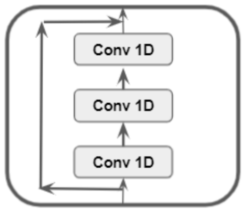
(1) Input Sequence
(2) Convolution 1D (filters: 16, kernel size: 4, activation: linear)

BuildSys '21, November 17–18, 2021, Coimbra, Portugal

ACM ISBN 978-1-4503-9114-6/21/11…$15.00
https://doi.org/10.1145/3486611.3486652



(a) Convolutional Block



(b) Identity Block

Figure 1: Two types of residual blocks containing skip connections are used for our ResNet model.

(3) BiLSTM (units: 128, activation: tanh)
(4) BiLSTM (units: 256, activation: tanh)
(5) Dense layer (units: 128, activation: tanh)
(6) Dense layer (units: 1, activation: linear)

2.2 BiLSTM with Attention Mechanism

LSTMs treat different parts of the input sequence "equally". Research literature has established the benefits of focusing the "attention" on specific parts of the input [1]. The next implementation is based on a recent paper [15] applying attention on top of the LSTMs.

BiLSTM Model with Attention

(1) Input Sequence
(2) Convolution 1D (filters: 16, kernel size: 4, activation: linear)
(3) BiLSTM (units: 128, activation: tanh)
(4) BiLSTM (units: 256, activation: tanh)
(5) Attention (units: 128)
(6) Dense layer (units: 128, activation: tanh)
(7) Dense layer (units: 1, activation: linear)

2.3 Residual Neural Networks (ResNet)

Residual Neural Networks (ResNet) [8, 9] are a combination of different types of residual blocks. These residual blocks contain skip connections. The skip connections add the output of the previous layer in the subsequent layers. Thus, skip connections preserve the output of the previous layer and solve the problem of vanishing gradients. Figure 1 shows the residual blocks used in the model.

ResNet model

(1) Input Sequence
(2) ZeroPadding1D (padding:3)
(3) BatchNormalization (axis:2)
(4) Activation (relu)
(5) Convolutional Block (filters: 30, kernel size: 24)
(6) Identity Block (filters: 30, kernel size: 12)
(7) Identity Block (filters: 30, kernel size: 6)
(8) Dense layer (units: 1024, activation: relu)
(9) Dropout (0.2)
(10) Dense layer (units:sequence length, activation:linear)

2.4 Classification Subnetwork

Multi-task learning [6] inspires the architecture of this model. In this model, there are two subnetworks, classification subnetwork and regression subnetwork. The classification subnetwork determines the on-off state of an appliance. The regression subnetwork will generate the value of appliance power consumption. The multiplication of both outputs is the final output as shown in Figure 2. As proposed in paper [15], the classification subnetwork will be as follows:

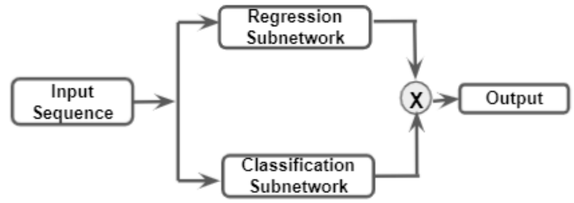


Figure 2: Basic structure of model with regression subnetwork and classification subnetwork. The classification subnetwork provides a path for accurately detecting on-off state of appliance. The final output is a product of regression and classification subnetwork.

Classification subnetwork

(1) Input Sequence
(2) Convolution 1D (filters: 30, kernel size: 10, activation: relu)
(3) Convolution 1D (filters: 30, kernel size: 8, activation: relu)
(4) Convolution 1D (filters: 40, kernel size: 6, activation: relu)
(5) Convolution 1D (filters: 50, kernel size: 5, activation: relu)
(6) Convolution 1D (filters: 50, kernel size: 5, activation: relu)
(7) Convolution 1D (filters: 50, kernel size: 5, activation: relu)
(8) Dense layer (units: 1024, activation: relu)
(9) Dense layer (units: sequence length, activation: sigmoid)

We have created two models with the same classification subnetwork as explained above and different regression subnetworks. In the first model, the regression subnetwork will be the same as the BiLSTM model with attention mechanism as explained in Section 2.2. In the second model, the regression subnetwork is similar to the ResNet model as explained in Section 2.3.

2.5 Bidirectional Encoder Representation from Transformers (BERT)

BERT has a local attention mechanism, due to which it considers part of a sequence for generating weights. Due to its feature of local attention, it was recently proposed for NILM [17]. As shown in Figure 3, the BERT model contains two important blocks: tokenizer and transformer block. The positional embedding in tokenizer adds the information about the position of the reading in the sequence. The transformer generates attention heads through the multi-head attention block. All the attention heads of the multi-attention layer are concatenated to get the single attention head. The output of the transformer block is passed through a dense layer to obtain the final output. In the interest of space, we do not provide the detailed blocks for BERT in the current paper and refer the reader to NILMTK-contrib repo.

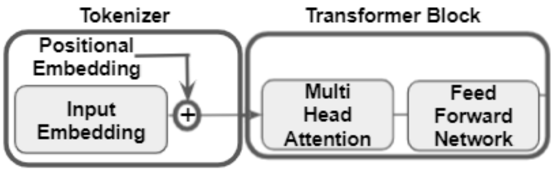


Figure 3: BERT model containing two important blocks: the transformer block and tokenizer. The importance of the tokenizer is to provide the positional information and transformer block contributes in calculating attention head.

3 PARSER FOR IDEAL DATASET

The IDEAL Household Energy Dataset [16] has appliance and mains energy data from 39 homes from the UK. One of our important motivation to create a NILMTK parser for this dataset was that is now one of the largest (in terms of homes and time) publicly available dataset[1]. Here, we introduced the data parser for the IDEAL dataset according to the NILM metadata [10]. We have pushed the data converter to the NILMTK repository.

4 EVALUATION

In this section, we describe the dataset, settings, benchmark algorithms, metrics and results. The main aim of the evaluation is to show the comparison between benchmark algorithms and the newly implemented algorithms.

4.1 Datasets

We have used the REDD dataset [13] and IDEAL dataset [16]. REDD has data from 6 homes over several weeks. The *mains* data is at one second frequency, and the appliance data is at three seconds frequency. In the IDEAL dataset, *mains* data is at one second frequency, and appliance data is at five seconds frequency.

4.2 Baseline Implementations

We use three baseline algorithms: Mean [12], Seq2Seq and Seq2point [18]. The Seq2Point algorithm has been verified earlier as the state-of-the-art algorithm [5].

4.3 Metrics

We have used mean absolute error (MAE)=$\frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|$ as our metric. Here, n is the number of readings, $\hat{y}_i$ is the predicted appliance reading, and y_i is the ground truth reading of an appliance.

4.4 Experimental Setup

For the REDD dataset and IDEAL dataset, Table 1 provides details regarding houses selected for training and testing. The selection of homes for the experiment was based on the availability of appliances, the contribution of appliance consumption on total consumption and the correlation in consumption between houses used for a particular appliance. Particularly for the IDEAL dataset, we have selected the homes with a relatively high amount of appliance data. We have resampled the *mains* reading and appliance power consumption for both the dataset at 60 seconds frequency. Our experiments and choice of appliances are heavily influenced by previous work [3, 4, 18]. We used leave-one-home-out cross-validation (where all but the test home are used for training) where the set of homes is mentioned in Table 1 for both the datasets. We have used a batch size of 32 and 50 epochs for training all the experiments. We used fine-tuned hyper-parameters such as input sequence length for different algorithms as proposed in the paper [5]. We have pushed all our experiments to NILMTK-contrib repository [2].

Table 1: Houses used for evaluation.

(a) REDD

Appliance	House No
Fridge	1,2,3,4,5
Dishwasher	1,2,3
Microwave	1,2,3
Washer Dryer	1,2

(b) IDEAL

Appliance	House No
Fridge	90, 136
Washing Machine	96, 136

4.5 Results and Analysis

The main result for our experiments can be found in Table 2. The performance for new algorithms is comparable or better than the Seq2Seq and Seq2point algorithms. For a subset of the appliances, the proposed algorithms are performing better. In the REDD dataset, the BiLSTM with attention layer and classification network, achieves the best performance for fridge. We believe that the presence of a classification network helps to detect the duty cycle accurately. The Seq2Seq and BiLSTM with attention layer algorithm are comparable for the dishwasher in terms of MAE, but as seen in Figure 4, Seq2Seq is predicting the appliance to be always off. In contrast, BiLSTM with attention is accurately generating the power consumption, though with some false positives. We believe that the improved performance of BiLSTM with attention weights is due to the focus on appropriate input corresponding to appliance usage. For microwave, the Seq2Seq and ResNet are comparable. The skip connections, as explained in Section 2.3 play a critical role in ensuring learnability in 'deeper' networks.

For the IDEAL dataset, we can see in Table 2, that the proposed algorithms perform better for both the appliances. For the fridge, BiLSTM with attention model performs best. For washing machine, the ResNet model is comparable with Seq2point.

4.6 Limitations and Future Work

The current work empirically compares neural methods on two datasets. The comparison will be more useful when done on a larger number of datasets across different countries. More importantly, in the future, we would like to answer the following two questions: i) "explaining" the good performance of newer models [14]; ii) quantifying under which situations a particular model works best.

4.7 Conclusions

In this paper, we have implemented five new neural network based models for NILM. We have even created a parser for the publicly

[1]The Dataport dataset from PecanStreet, which was earlier the largest publicly available dataset requires a paid license now.

[2]https://github.com/nilmtk/nilmtk-contrib

Table 2: MAE (lower is better) on the REDD dataset and IDEAL dataset after cross-validation. The newly implemented algorithms performed better for subset of appliances in comparison to the baseline implementation.

Algorithms	REDD				IDEAL	
	Fridge	Dishwasher	Microwave	Washer Dryer	Fridge	Washing Machine
Seq2Point	33.84	14.43	16.11	**38.06**	27.02	33.64
Seq2Seq	31.33	13.98	**14.05**	44.13	23.98	36.25
Mean	75.15	22.11	19.68	124.49	31.71	47.29
BiLSTM	33.69	15.62	19.62	48.01	17.07	41.29
BiLSTM + Attention	36.05	**13.84**	21.14	63.31	**16.96**	37.84
BiLSTM + Attention + Classification	**26.73**	20.08	18.94	59.89	18.95	40.10
BERT	43.12	21.51	20.71	50.46	25.46	57.77
Resnet	33.34	14.29	14.09	55.33	23.99	**33.17**
Resnet + Classification	27.07	20.24	18.18	48.07	19.80	34.68

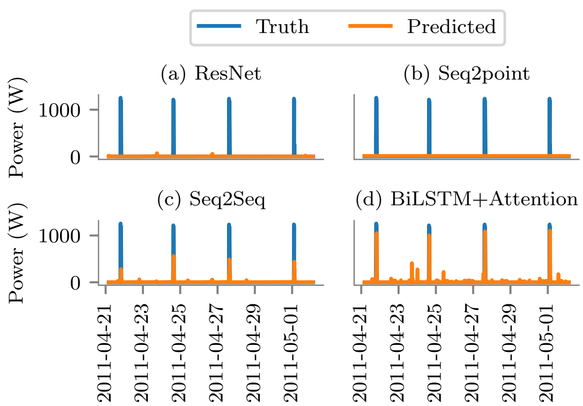


Figure 4: Four algorithms resulted in almost comparable MAE for dishwasher. (a). ResNet predicts constant off-state. (b). Seq2point predicts constant off-state. (c). Seq2Seq is able to detect the on states but cannot estimate the power consumption accurately. (d). BiLSTM with attention generates values accurately due to attention mechanism.

available dataset. In addition to this, we have provided a comparative study of the benchmark algorithms with five newly implemented ones. The newly proposed algorithms beat the state-of-the-art NILM methods and pave the way forward for NILM research.

5 ACKNOWLEDGEMENTS

We would like to thank Cisco Research for sponsoring this research. We would like to thank KSVR Rithwik and Zeel Patel from IIT Gandhinagar for their inputs.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).
[2] Nipun Batra, Jack Kelly, Oliver Parson, Haimonti Dutta, William Knottenbelt, Alex Rogers, Amarjeet Singh, and Mani Srivastava. 2014. NILMTK. *Proceedings of the 5th international conference on Future energy systems* (Jun 2014). https://doi.org/10.1145/2602044.2602051
[3] Nipun Batra, Jack Kelly, Oliver Parson, Haimonti Dutta, William Knottenbelt, Alex Rogers, Amarjeet Singh, and Mani Srivastava. 2014. NILMTK: an open source toolkit for non-intrusive load monitoring. In *Proceedings of the 5th international conference on Future energy systems.* 265–276.
[4] Nipun Batra, Rithwik Kukunuri, Ayush Pandey, Raktim Malakar, Rajat Kumar, Odysseas Krystalakos, Mingjun Zhong, Paulo Meira, and Oliver Parson. 2019. 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.* 193–202.
[5] Nipun Batra, Rithwik Kukunuri, Ayush Pandey, Raktim Malakar, Rajat Kumar, Odysseas Krystalakos, Mingjun Zhong, Paulo C. M. Meira, and Oliver Parson. 2019. Towards reproducible state-of-the-art energy disaggregation. *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation* (2019).
[6] Anthony Faustine, Lucas Pereira, Hafsa Bousbiat, and Shridhar Kulkarni. 2020. 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.* 84–88.
[7] George William Hart. 1992. Nonintrusive appliance load monitoring. *Proc. IEEE* 80, 12 (1992), 1870–1891.
[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03385 [cs.CV]
[9] Ziyue Jia, Linfeng Yang, Zhenrong Zhang, Hui Liu, and Fannie Kong. 2020. Sequence to Point Learning Based on Bidirectional Dilated Residual Network for Non Intrusive Load Monitoring. arXiv:2006.00250 [eess.SP]
[10] Jack Kelly and William Knottenbelt. 2014. Metadata for Energy Disaggregation. In *The 2nd IEEE International Workshop on Consumer Devices and Systems (CDS 2014).* Västerås, Sweden. arXiv:1403.5946
[11] Jack Kelly and W. Knottenbelt. 2015. Neural NILM: Deep Neural Networks Applied to Energy Disaggregation. *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments* (2015).
[12] J Zico Kolter, Siddharth Batra, and Andrew Y Ng. 2010. Energy disaggregation via discriminative sparse coding. In *Advances in Neural Information Processing Systems.* 1153–1161.
[13] J. Zico Kolter and Matthew J. Johnson. 2011. REDD: A Public Data Set for Energy Disaggregation Research. In *IN SUSTKDD.*
[14] David Murray, Lina Stankovic, and Vladimir Stankovic. 2020. Explainable nilm networks. In *Proceedings of the 5th International Workshop on Non-Intrusive Load Monitoring.* 64–69.
[15] Veronica Piccialli and Antonio M. Sudoso. 2021. Improving Non-Intrusive Load Disaggregation through an Attention-Based Deep Neural Network. *Energies* 14, 4 (Feb 2021), 847. https://doi.org/10.3390/en14040847
[16] Martin Pullinger, Jonathan Kilgour, Nigel Goddard, Niklas Berliner, Lynda Webb, Myroslava Dzikovska, Heather Lovell, Janek Mann, Charles Sutton, Janette Webb, et al. 2021. The IDEAL household energy dataset, electricity, gas, contextual sensor data and survey data for 255 UK homes. *Scientific Data* 8, 1 (2021), 1–18.
[17] Zhenrui Yue, Camilo Requena Witzig, Daniel Jorde, and Hans-Arno Jacobsen. 2020. BERT4NILM: A Bidirectional Transformer Model for Non-Intrusive Load Monitoring. In *Proceedings of the 5th International Workshop on Non-Intrusive Load Monitoring.* 89–93.
[18] Chaoyun Zhang, Mingjun Zhong, Zongzuo Wang, Nigel Goddard, and Charles Sutton. 2018. Sequence-to-point learning with neural networks for non-intrusive load monitoring. In *Thirty-second AAAI conference on artificial intelligence.*