

Demo Abstract: A demonstration of reproducible state-of-the-art energy disaggregation using NILMTK

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

Raktim Malakar
IEST Shibpur
raktimalakar2015@gmail.com

Mingjun Zhong
University of Lincoln
MZhong@lincoln.ac.uk

Rithwik Kukunuri
IIT Gandhinagar
kukunuri.sai@iitgn.ac.in

Rajat Kumar
DAIICT
201811024@daiict.ac.in

Paulo Meira
University of Campinas
pmeira@ieee.org

Ayush Pandey
NIT Agartala
ayush20pandey@gmail.com

Odyseas Krystalakos
Aristotle University of Thessaloniki
odyseaskrst@gmail.com

Oliver Parson
Centrica Hive Limited
oliver.parson@hivehome.com

ABSTRACT

Non-intrusive load monitoring (NILM) or energy disaggregation involves separating the household energy measured at the aggregate level into constituent appliances. The NILM toolkit (NILMTK) was introduced in 2014 towards making NILM research reproducible. NILMTK has served as the reference library for data set parsers and reference benchmark algorithm implementations. However, few publications presenting algorithmic contributions within the field went on to contribute implementations back to the toolkit. This work presents a demonstration of a new version of NILMTK [2] which has a rewrite of the disaggregation API and a new experiment API which lower the barrier to entry for algorithm developers and simplify the definition of algorithm comparison experiments. This demo also marks the release of NILMTK-contrib: a new repository containing NILMTK-compatible implementations of 3 benchmarks and 9 recent disaggregation algorithms. The demonstration covers an extensive empirical evaluation using a number of publicly available data sets across three important experiment scenarios to showcase the ease of performing reproducible research in NILMTK.

CCS CONCEPTS

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

KEYWORDS

energy disaggregation; non-intrusive load monitoring; smart meters

ACM Reference Format:

Nipun Batra, Rithwik Kukunuri, Ayush Pandey, Raktim Malakar, Rajat Kumar, Odyseas Krystalakos, Mingjun Zhong, Paulo Meira, and Oliver Parson. 2019. Demo Abstract: A demonstration of reproducible state-of-the-art energy disaggregation using NILMTK. In *The 6th ACM International Conference*

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

BuildSys '19, November 13–14, 2019, New York, NY, USA

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-7005-9/19/11...\$15.00

<https://doi.org/10.1145/3360322.3360999>

on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '19), November 13–14, 2019, New York, NY, USA. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3360322.3360999>

1 INTRODUCTION

Non-intrusive load monitoring (NILM) or energy disaggregation is the task of separating a building's energy measured at the aggregate level into constituent appliances. The problem was originally studied by Hart in the early 1980s [3] and has seen a renewed interest in recent years. The open source non-intrusive load monitoring toolkit (NILMTK) [1] was released in 2014 to enable easy comparison of NILM algorithms in a reproducible fashion. The main contributions of the toolkit were: i) NILMTK-DF (data format): the standard energy disaggregation data structure used by NILMTK; ii) parsers for six existing data sets; iii) implementations of two benchmark NILM algorithms; iv) statistical and diagnostic functions for understanding data sets; v) a suite of accuracy metrics across a range of use cases. Later in 2014, NILMTK v0.2 was released [4] which added support for out-of-core computation, motivated by release of very large data sets such as Pecan Street Inc's Dataport data set [8].

Since these two releases, NILMTK has become the energy disaggregation field's reference library for data set parsers and reference benchmark algorithm implementations. However, few publications presenting algorithmic contributions within the field went on to contribute implementations back to the toolkit. This paper and demo will highlight two significant contributions to the NILM research community: i) a rewrite of the disaggregation API and implementation of a new experiment API, which respectively lower the barrier to entry for algorithm developers and simplify the definition of algorithm comparison experiments, ii) release of NILMTK-contrib: a new repository containing NILMTK-compatible implementations of 3 benchmarks and 9 recently published disaggregation algorithms [5–7, 9, 10]. Algorithm developers will be able to compare the performance of a new approach with state-of-the-art algorithms in a range of different settings.

We demonstrate the versatility of the new experiment API by conducting experiments across the following 3 train/test scenarios: i) train and test on a different building from the same data set, ii) train on multiple buildings from different data sets and iii) train and test on artificially generated aggregate data (by summing appliance

usage rather than measuring the building aggregate). Furthermore, we demonstrate the potential of NILMTK-contrib by comparing the performance of the 3 benchmarks and 9 disaggregation algorithms included in NILMTK-contrib in each of these settings.

2 IMPROVEMENTS TO NILMTK

Two core changes have been made to NILMTK; a new experiment interface and a rewrite of the model interface; and various user-raised issues have been addressed.

Experiment Interface: This release introduces ExperimentAPI; a new NILMTK interface which reduces the barrier-to-entry for specifying experiments for NILM research. The new interface allows NILMTK users to focus on which experiments to run rather than on the code required to run such experiments. Our new interface decouples *what we want to do* (declarative) from *how we do it* (imperative). The new interface encapsulates parameters required for training and testing over data sets using NILMTK, which were previously spread across multiple NILMTK modules. Thus, the new API drastically reduces the workload for the toolkit user.

Model Interface: The disaggregator class in the previous version required intricate knowledge of NILMTK objects, which proved to be a barrier for community algorithm authors. We thus introduced a new Model Interface where the class definition has been simplified in terms of input and output formats and is consistent throughout the new API, but also all of the new functions in are independent of NILMTK objects. The algorithm developer only needs to know the PyData stack to write new NILMTK algorithms.

Practical Improvements to NILMTK: Most of the 727 NILMTK Github issues have been fixed. Addressing these issues and improving NILMTK involved work from 21 contributors. 1700+ commits and 72 pull requests later, 617 issues have been closed as of now. Our main efforts have spanned towards: 1) *Simplifying installation; Introducing new data converters and improvements;* iii) *Significant documentation increase*

3 NILMTK-CONTRIB

We have chosen to house bleeding-edge algorithms in a separate repository to the core toolkit to encourage algorithm publishers to own the implementation of their algorithm. We expect algorithms to eventually move into the main NILMTK repository. The following algorithms mentioned in this section have been implemented in accordance with the new disaggregator class described earlier: **i) Mean:** The Mean algorithm is a simple benchmark designed to provide a well-understood baseline against which more complex algorithms can be compared; **ii) Edge Detection:** Proposed by George Hart [3], this algorithm is often used as a baseline model for the NILM problem. The technique is based on edge detection within the power signal, which divides the time series into steady and transient time periods; **iii) Combinatorial Optimisation (CO):** The goal of the algorithm is to assign states to appliances in a way that the difference between the household aggregate reading and the sum of power usage of the different appliances is minimised; **iv) Discriminative Sparse Coding (DSC):** Sparse coding approximates the original energy matrix by representing it as a product of over-complete bases and their activations [6]; **v) Exact Factorial Hidden Markov Model (ExactFHMM):** In an FHMM,

each appliance is represented by a hidden Markov model; **vi) Approximate Factorial Hidden Markov Model (ApproxFHMM)** Inference of exact solutions in an FHMM is expensive and often becomes stuck in a local optimum. The approximate factorial hidden Markov model aims to alleviate these issues by relaxing the state values into $[0, 1]$ and transforming the FHMM inference problem to a convex program [10]. **vii) FHMM with Signal Aggregate Constraints (FHMM+SAC):** FHMM + SAC is an extension to the baseline FHMM, where the aggregate value of each appliance over a time period is expected to be a certain value [10]; **viii) Denoising Autoencoder (DAE):** The denoising autoencoder is a neural network architecture designed to extract a particular component (appliance power) from noisy input (mains aggregate) [5]; **ix) Recurrent Neural Network (RNN):** Kelly et al. proposed a RNN that receives a sequence of mains readings and outputs a single value of power consumption of the target appliance [5]. To overcome the vanishing gradient problem, the network utilises long short-term memory (LSTM) units; **x) Sequence-to-Sequence (Seq2seq)** The sequence to sequence learning model [9] learns a regression map from the mains sequence to the corresponding target appliance sequence; **xi) Sequence-to-Point (Seq2point)** Following the work in [9], sequence to point learning (seq2point) models the input of the network as a mains window and the output as the midpoint element of the corresponding window of the target appliance; **xii) Online GRU:** Krystalakos et al. proposed a GRU architecture that attempts to reduce the computational demand while maintaining the same performance [7].

REFERENCES

- [1] 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*. ACM, 265–276.
- [2] Nipun Batra, Rithwik Kukuluri, 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 Embedded Systems for Energy-Efficient Built Environments (BuildSys '19)*. ACM, New York, NY, USA.
- [3] G. W. Hart. 1992. Nonintrusive appliance load monitoring. *Proc. IEEE* 80, 12 (Dec. 1992), 1870–1891. <https://doi.org/10.1109/5.192069>
- [4] Jack Kelly, Nipun Batra, Oliver Parson, Haimonti Dutta, William Knottenbelt, Alex Rogers, Amarjeet Singh, and Mani Srivastava. 2014. Nilmtk v0. 2: a non-intrusive load monitoring toolkit for large scale data sets: demo abstract. In *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-efficient Buildings*. ACM, 182–183.
- [5] Jack Kelly and William Knottenbelt. 2015. Neural NILM: Deep Neural Networks Applied to Energy Disaggregation. In *Proceedings of the 2Nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments (BuildSys '15)*. ACM, New York, NY, USA, 55–64. <https://doi.org/10.1145/2821650.2821672> event-place: Seoul, South Korea.
- [6] J. Z. Kolter, Siddharth Batra, and Andrew Y. Ng. 2010. Energy Disaggregation via Discriminative Sparse Coding. In *Advances in Neural Information Processing Systems 23*, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta (Eds.), Curran Associates, Inc., 1153–1161. <http://papers.nips.cc/paper/4054-energy-disaggregation-via-discriminative-sparse-coding.pdf>
- [7] Odysseas Krystalakos, Christoforos Nalmpantis, and Dimitris Vrakas. 2018. Sliding Window Approach for Online Energy Disaggregation Using Artificial Neural Networks. In *Proceedings of the 10th Hellenic Conference on Artificial Intelligence*.
- [8] Oliver Parson, Grant Fisher, April Hersey, Nipun Batra, Jack Kelly, Amarjeet Singh, William Knottenbelt, and Alex Rogers. 2015. Dataport and nilmtk: A building data set designed for non-intrusive load monitoring. In *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 210–214.
- [9] 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*. <https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16623>
- [10] Mingjun Zhong, Nigel Goddard, and Charles Sutton. 2014. Signal Aggregate Constraints in Additive Factorial HMMs, with Application to Energy Disaggregation. In *Advances in Neural Information Processing Systems 27*.