Poster: Introducing MILM - A Hybrid Minimal-Intrusive Load Monitoring Approach

Benjamin Völker,
Marc Pfeifer
[voelkerb,pfeiferm]@informatik.uni-feiburg.de
Computer Architecture
University of Freiburg
Germany

Florian Wolling
florian.wolling@uni-siegen.de
Ubiquitous Computing
University of Siegen
Germany

Philipp M. Scholl
pscholl@informatik.uni-freiburg.de
Computer Architecture
University of Freiburg
Germany

Bernd Becker
becker@informatik.uni-freiburg.de
Computer Architecture
University of Freiburg
Germany

ABSTRACT

The shift towards an advanced electricity metering infrastructure has gained traction because of several smart meter roll-outs. This accelerated research in Non-Intrusive Load Monitoring techniques. These techniques highly benefit from the temporal resolution improvements achieved by smart meters. Nevertheless, industrial adoption is low, not least because the achieved disaggregation performance is rather poor for unsupervised approaches. This work sketches a way to utilize intrusive sensors in combination with a standard NILM system to enhance training and maximize overall system's performance while minimizing the number of required intrusive sensors.

CCS CONCEPTS

• **Human-centered computing** → *HCI theory, concepts and models*; • **Hardware** → *Power and energy*; • **Computing methodologies** → *Classification and regression trees.*

KEYWORDS

Minimal-Intrusive Load Monitoring, NILM, appliance classification

ACM Reference Format:
Benjamin Völker, Marc Pfeifer, Florian Wolling, Philipp M. Scholl, and Bernd Becker. 2021. Poster: Introducing MILM - A Hybrid Minimal-Intrusive Load Monitoring Approach. In *The Twelfth ACM International Conference on Future Energy Systems (e-Energy '21), June 28–July 2, 2021, Virtual Event, Italy*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3447555.3466578

1 INTRODUCTION

Reducing our electricity consumption is a vital step to achieve the goal of saving the earth's energy resources. In the residential domain energy monitoring and 'eco-feedback' techniques have proven to help by raising the awareness of unnecessary electricity consumption. According to several studies (e.g. [2]) such feedback is particularly effective (5–10 % relative increase), if the feedback is provided with individual appliance consumption data.

Non-Intrusive Load Monitoring (NILM) is a way to get appliance-specific consumption data without replacing existing appliance or infrastructure. A single electricity meter measures the composite load and disaggregates it into the load of each consumer. Despite being researched for over three decades, NILM methods still suffer from mediocre disaggregation performance [1] if not specifically trained on the target home in a supervised fashion. Such training requires home owners to switch appliances *on* and *off* multiple times with different combinations of concurrently running devices, taking the word 'non-intrusive' *ad absurdum*. In this work, we investigate ways to aid and accelerate the training process for supervised NILM systems by using (intrusive) plug meters. We present a system which uses these meters in an adaptive training approach, which determines a minimal set of meters required to boost overall system performance to a desired level before prompting the home owner to change the meter configuration. We term this approach Minimal-Intrusive Load Monitoring (MILM) to hint towards the fact that the training process is optimized towards a minimal and practical effort.

2 MINIMAL INTRUSIVE LOAD MONITORING

In a MILM system, intrusive plug meters monitor an individual appliances' erngy consumption in addition to a central smart meter for monitoring a powerline's aggregated load, as shown in Figure 1. These intrusive plug meters are used for (1) the collection of training samples not impeded by noise, (2) the collection of ground truth labels for the aggregated data, and to (3) aid the training process for home owners. We call this concept Minimal-Intrusive Load Monitoring (MILM) as it combines the advantages of NILM with the advantage of individual metering. We consider it to be 'minimal' intrusive since the amount of individual meters is limited during training, and minimized after. Deploying the system in a new home requires to install an aggregating meter (e.g. a smart meter) into the home's fuse box and a handful of plug meters ($m_0, \ldots, m_N$) are handed to the home owner. For simplicity let us consider only a single plug meter is used. The home owner decides which appliance (a_j with $j \in [0, \ldots M]$) he is interested in most and installs a plug meter to it. Using an event detection algorithm the system collects an event set (ES_j) for appliance a_j. This set consists of the events of a_j taken from the plug meter and the smart meter as cleaned aggregated data. Both events are labeled as *POS*. As both data are included in the training set, we refer to this as *hybrid training*.

e-Energy '21, June 28–July 2, 2021, Virtual Event, Italy

ACM ISBN 978-1-4503-8333-2/21/06…$15.00
https://doi.org/10.1145/3447555.3466578

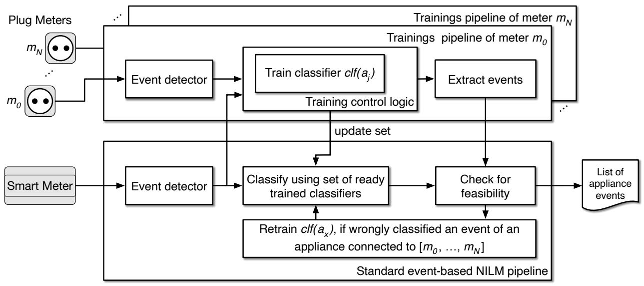

Figure 1: Concept of Minimal-Intrusive Load Monitoring with a set of N intrusive meters. Plug meters are consecutively connected to an appliances of interest a_j guided by a user-centric training phase.

Furthermore, all other events found in the aggregated data are added to ES_j labeled as *NEG*. If sufficient *POS* and *NEG* events have been added, a binary classifier is trained on ES_j by splitting the set into 20% test samples ($TEST_j$) and 80% training samples ($TRAIN_j$). All individual appliance data is removed from $TEST_j$ to ensure that the classifier is tested on unseen appliance events solely from the aggregated data. This simulates the case that the plug meter was removed from the appliance, and is monitored only via the aggregated measurements. A classifier (e.g. an SVM) is trained on $TRAIN_j$. For each new positive sample added to ES_j, the training is repeated. Training stops, if the classifier has reached a desired F_1-score $F_{1\ max}$ for multiple training runs (*success*) or if the learning does not improve anymore (*failure*). On *success*, the user is notified that training for a_j has finished and asks the user to connect plug meter m_i to another appliance (a_{j+1}) for which the process is repeated. On *failure* events of a_j cannot sufficiently be recognized by the system using aggregated data only. The user is notified and requested to leave m_i installed if he wants to continue to meter the events of this appliance. All events in the aggregated data triggered by such an appliance are not added to the training sets of other appliances and solely the intrusive meter is used to determine events of a_j - hence cleaning the aggregated data from possibly confounding data. Each successfully trained classifier $clf(a_j)$ is added to a set of classifiers (*CF*) and the final training set $TRAIN_j$ is stored. *CF* is used in a one-vs-rest strategy to achieve multi-class classification of all appliances in *CF*, i.e. the classifier with the highest confidence determines the assigned class. If no classifier estimates a confidence of more than 50% for the *POS* class, the event is regarded as *unknown.* We further utilize the events labeled by each m_i to test all classifiers (depicted in Figure 1 as *Check for feasibility*):

- If a currently trained classifier $clf(a_j)$ mistakenly classifies an event as an event of a_j (*false positive*), the event is added to an *explicit negative*-class and a retraining of $clf(a_j)$ is scheduled.
- If a currently trained classifier $clf(a_j)$ misses an aggregated event of a_j (*false negative*) it is retrained immediately.
- If a classifier $clf(a_x)$ in *CF* mistakenly classifies an event known to be an event of a currently metered appliance (*false positive*), it is also added to $TRAIN_x$ as an *explicit negative*-class and a retraining of $clf(a_x)$ is scheduled.

Furthermore, all extracted events of each individually metered appliance are added to the event list, which allows to monitor for appliance events in case of *failure* and during the training phase. Therewith, the proposed MILM system can determine the events of all appliances of interest (i.e. timestamp and appliance causing the event). This is achieved by using the combination of a NILM system with a small number of (intrusive) plug meters to determine a final minimal required number of plug meter for a given detection performance.

3 EVALUATION ON FIRED

We evaluated the approach on 71 days of the FIRED [3] dataset. In these preliminary experiments we varied four different $F_{1\ max}$-thresholds, two connection sequences, and we assumed a total of five plug meters. The connection sequence is ordered by (1) the average energy consumption (CS_{power}) and (2) the number of events exhibited (CS_{events}) by an appliance. An appliance contributes to the overall performance as soon as its training has finished (*success*). Table 1 shows the results using an Support Vector Machine (SVM) classifier. It can be seen that a maximum of ten appliances were learned within the given time period. This seems to be a comparably small portion of all 19 appliances. However, most appliances did not produce enough events in the given time period. Hence, it is generally more beneficial in terms of the system's learning rate to attach appliances by their exhibited number of events (CS_{events}). Lowering $F_{1\ max}$ generally leads to a faster learning rate but also to a worse F_1-score. The best results are achieved for CS_{events} and $F_{1\ max}$ = 0.9. While the F_1-score is lower compared to $F_{1\ max}$ = 0.95, the number of learned appliances is significantly higher. The system does not perform disaggregation into load profiles yet. It rather delivers a set of appliance events from which appliance *on*-phases can be identified. Disaggregation into load profiles remains a challenge for future work.

Table 1: Evaluation of MILM on FIRED. f and s represent the number of appliances for learning *failed* and *succeeded.*

	CS_{events}						CS_{power}					
$F_{1\ max}$	f	s	Pr	Re	Ac	F_1	f	s	Pr	Re	Ac	F_1
0.80	0	9	0.89	0.93	0.97	0.91	0	5	0.87	0.93	0.97	0.91
0.85	0	9	0.89	0.93	0.97	0.91	0	4	0.89	0.89	0.97	0.9
0.90	0	10	0.92	0.92	0.97	0.92	1	3	0.94	0.97	0.97	0.97
0.95	4	2	0.99	0.99	0.99	**0.99**	2	2	0.99	1.0	0.98	**0.99**

4 CONCLUSION

In this work, we introduced MILM which uses intrusive meters to generate labeled hybrid appliance data which renders load disaggregation practical for new environments.

REFERENCES

[1] A. Reinhardt and C. Klemenjak. 2020. How does load disaggregation performance depend on data characteristics? insights from a benchmarking study. In *Proceedings of the 11th ACM International Conference on Future Energy Systems.* 167–177.

[2] Tiago Serrenho, Paolo Zangheri, and Paolo Bertoldi. 2015. *Energy Feedback Systems: Evaluation of Meta-studies on energy savings through feedback.* Technical Report EUR 27992 EN. Publications Office of the European Union. doi: 10.2790/565532.

[3] Benjamin Völker, Marc Pfeifer, Philipp M Scholl, and Bernd Becker. 2020. FIRED: A Fully-labeled hIgh-fRequency Electricity Disaggregation Dataset. In *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation.* 294–297.