Using the Wisdom of Neighbors for Energy Disaggregation from Smart Meters

Nikolay Laptev, Yuting Ji, and Ram Rajagopal

***Abstract*—The problem of disaggregating the household electricity demand into appliance-level consumption is considered. A deep neural network, based on the long short term memory model, is developed to jointly predict all appliances by leveraging information of neighboring homes. The proposed technique is evaluated on a real-world data set with over 300 homes and more than 20 appliances. Numerical results show significant improvement of the proposed technique over the baseline without joint training and using the wisdom of neighbors.**

***Index Terms*—Deep learning, energy disaggregation, long short term memory, non-intrusive load monitoring (NILM).**

I. Introduction

Non-intrusive load monitoring (NILM) separates the whole building or home (aggregate) energy data into appliance specific data. With the widespread deployment of advanced metering infrastructure, NILM draws attention from industry and academia at unprecedented levels.

Energy disaggregation has numerous benefits. For customers, an itemized electricity bill help them to better understand their home's energy use which may in turn lead to consumption reduction. According to [1], disaggregated energy feedback leads to electricity reduction ranging from 2% to 18% in trial studies or pilot programs. For grid operators, disaggregation may improve the demand prediction because fine-grained models and algorithms are readily to use. For utility companies, disaggregated energy data allows better user segmentation and targeted demand response. In particular, energy efficient programs at the appliance level, such as SmartAC™ program of Pacific Gas and Electric Company, On Call program of Florida Power and Light Company, and EV 360™ of Austin Energy, require energy data for specific appliances.

Accurate energy disaggregation, however, is challenging. First, small loads such as kitchen appliances and lights get lost in the noise of large loads like AC and electric car. Second, most existing approaches to disaggregation require a model of each appliance, and a manual process is required to generate or train these models. However, such models may not generalize to unseen homes. This is a key limitation to NILM because we can not collect training data for all homes of interest. Inference is essential to disaggregate energy data at large scale.

This work was supported in part by NSF under award ECCS-1554178, in part by DOE SunShot Office under award DE-EE00031003, and in part by DOE ARPA-E under award DE-AR0000697.

The authors are with the Department of Civil and Environmental Engineering, Stanford University, CA, 94305, USA (e-mail: nlaptev@stanford.edu; yutingji@stanford.edu; ramr@stanford.edu).

Relatively recently, time series modeling based on the Long Short Term Memory (LSTM) [2] model has gained popularity due to its end-to-end modeling, ease of incorporating exogenous variables, and automatic feature extraction abilities [3]. By providing a large amount of data across numerous dimensions, it has been shown that an LSTM network can model complex nonlinear feature interactions. A recent paper [4] has shown that a neural network forecasting model is able to outperform classical time series methods in cases with long, interdependent time series.

We propose an approach for time-series modeling using neural nets based on a multidimensional input to predict a multidimensional output. Under the conditions of noisy data, multiple features can be used to improve modeling accuracy. In our use-case of NILM, we use surrounding neighbor information as additional time-series signals to infer the disaggregate appliance usage. Furthermore, we employ joint training on a multidimensional output to leverage the correlation among home appliances.

This paper makes the following contributions:

- We show that using multiple related signals (aggregate energy data from neighboring homes) as input to the neural network model results in a lower disaggregation error for NILM applications.
- A reduction in disaggregation error is also achieved through the joint learning on the appliance output where a single model is constructed to predict power consumption of all appliances of interest at once.
- On evaluation, we test energy disaggregation algorithm on real-world data set with several hundreds homes and over twenty appliances.

II. Related Work

The early study of energy disaggregation dates back to 1990s [5]. Since then, various approaches have been explored and different techniques including statistical modeling [6], machine learning [7], and deep learning [8], have been applied. Traditional model-based approaches [5][6] usually assume a statistical or physical model for each appliance. Such models are difficult to justify its accuracy and generalize across homes. Learning based approaches [8][9] seem to be a more desirable alternative. Existing approaches in this category simply learn and disaggregate the signal of a particular appliance. Inter-relationship across appliances has not been studied in NILM. The proposed technique complements the existing learning algorithms. For a broader related work, see [10][11][12][13][14] and references therein. Below, we just discuss approaches most relevant to the technique developed here.

The authors of [8] are among the first to investigate deep learning techniques on NILM application. In particular, three deep neural network architectures are adapted to energy disaggregation with *one network per target appliance*. This vanilla version is different from the proposed approach in two aspects. First, we use an integrated neural network for all appliances to capture the inherent correlation. Second, information from other homes is also incorporated. Experimental results show significant improvement by utilizing the information across appliances and homes.

A particular relevant prior work is [9] where the authors incorporate data from neighbors who have similar consumption pattern in the energy disaggregation technique. Specifically, for each target home, the consumption break downs at appliance level are estimated by the *average* of its K nearest neighbors in the sense of *consumption similarity*. As the authors pointed out, the key intuition behind this approach is the hypothesis that consumption similarity at household-level might infer that at appliance-level. Our experiment results, however, show the opposite: neighbors with *similar* household consumption pattern leads to higher disaggregation error compared with randomly selected neighbors. In addition, the reason to use linear combination (average) of appliance consumption across similar neighbors is not provided in [9]. The proposed approach in this paper, on the other hand, does not assume the linear relation between neighbors for inferring the target home.

III. Learning Across Time-Series

Time-series data can be decomposed into the summation of a (usually monotonic) trend component, a (periodic) seasonal component, a holiday component, a stationary (stochastic) component, and a (usually i.i.d.) error component. Most of the time-series forecasting models assume explicit models with hyper-parameters of the trend, seasonality, and the stochastic process:

$$y(t) = g(t;\alpha_i^t) + s(t;\alpha_i^s) + h(t;\alpha_i^h) + c(t;\alpha_i^c) + \varepsilon(t;\alpha_i^\varepsilon),$$

where $g(\cdot)$, $s(\cdot)$, $h(\cdot)$, $c(\cdot)$, $\varepsilon(\cdot)$ represent the trend, seasonal, holiday, stationary, and error component, respectively.

These models usually use maximum likelihood estimation to determine the hyper-parameter vector α. However, the explicit model formulation of each component is still far from empirical understanding. For example, the authors of [15] propose to use only a piecewise-linear function and a logistic growth model for trending modeling, and a Fourier series with trigonometric function as basis for seasonality modeling. Moreover, all hyper-parameters need to be optimized for specific data sets.

With an artificial neural network (ANN), our aim is to eliminate the need to define the model explicitly. ANN is a graph of artificial neurons that allow information to pass to and from the neurons. Neurons are connected across layers and the edges between neurons have weights W. By modifying W the network learns. ANN has a *forward pass*, where the information flows from input to output and a *backward pass* during which the network updates W.

The LSTM model is a type of ANN which is applicable to temporal data. LSTM consists of 5 different nonlinear components. Initially, the LSTM cell is designed to repeat infinitely, with a single set of hyper-parameters, including four rectangular weight matrices W_f, W_h, W_u, W_o, acting on input vector x, serving for computing forget gate, candidate state, update gate, and output gate, respectively. Four square weight matrices R_f, R_h, R_u, R_o, acting on lagged output vector y_{t-1}, serving for the same computation procedures. An excellent visualization of how an LSTM works with a mathematical formulation is available in [16]. By stacking LSTM cells to construct a deep LSTM model with multiple layers, we gain additional freedom by enabling different weight matrices in different LSTM layers.

We hypothesize that the feature layers of LSTM model is a generalization of nonlinear cross correlation among the time-series that are given as input. Furthermore we hypothesize that it learns a cross correlation between individual time-series features such as the trend, seasonality, and holiday representation, in analogy to the explicit models in traditional time-series modeling tasks. While the model representation itself rather a general feature for all time-series data, we anticipate that the representation learned across the many time-series can be used for improving short term time-series forecasting accuracy.

In this paper we consider the neural network model architecture see Fig. 1) targeted for NILM applications with and without using the neighbor information. The former receives multiple time-series from the neighbors as input and provides a disaggregated appliance usage information for the target household for all appliances, while the latter only receives the historical target time-series as an input and provides the disaggregated appliance electricity usage per appliance requiring the construction of multiple models for each appliance as was done in previous work [8]. By training the network to output all appliances, we take advantage of joint output training.

During training the input time-series is normalized by the median of the current sliding window. Note that we use the same normalizing median of the target household aggregate power usage when normalizing the input time-series of other neighbors. Compared to [1] where a single source was used as input, we found that this normalization for multiple input signals was critical for our network to achieve best performance. Furthermore we found that training a single model to output all disaggregated appliances performed compared to a single model per appliance [1].

IV. Results

Dataset: The data set [17] used in this study is the hourly measurement of power consumption of homes and individual appliances. Data is collected from 345 homes with each having complete record for at least one appliance, mainly located in Austin, Texas, in 2016.

Deep LSTM Model: Our deep LSTM model consists of 6 LSTM layers and has a form of an autoencoder with a bottleneck layer. The first two LSTM layers have a 128-dimensional state vector, the bottlekneck layer has a 32 state

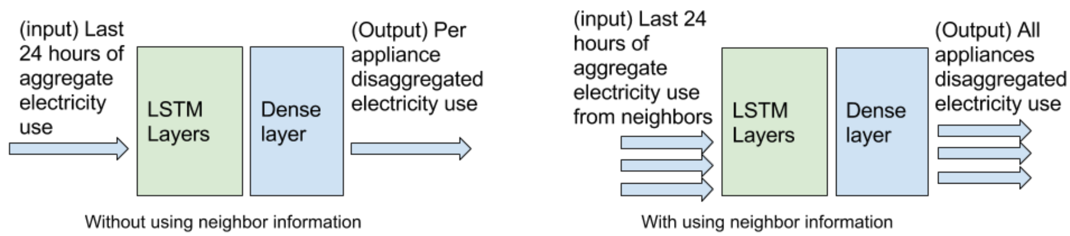

Fig. 1. Neural network model architecture with and without using the neighbor information.

vector and the last two layers have a 128-dimensional state vector. For a given timestamp t (hour), the input vector $\mathbf{x}$ consists of 12-hour history: $\mathbf{x} = [l_t, \cdots, l_{t-23}]$, and the output vector $\mathbf{y}$ consists of 12-hour electricity load from time t: $\mathbf{y} = [l_{t+1}, \cdots, l_{t+24}]$. We train the network using 12 hours for the forecast horizon and 12 hours for the lookback (for both usage data and appliance data).

By modeling the problem with an auto encoder, we are attempting to reconstruct a clean signal from a noisy one via a dimensionality reduction (e.g., bottleneck layer). In the image domain, a similar method was used for image denoising. In our case, a noisy image is equivalent to a multidimensional aggregate power usage and a clean image is equivalent to a disaggregated power demand per appliance for each household.

Methodology: The goal of our paper is to answer the question of the neural network can learn patterns across appliances from the same user and across users to aid the overall electricity usage prediction for appliance disaggregation use-case. Our hypothesis is that the network will be able to learn patterns across time-series (e.g., utilizing aggregation information from neighbors) to aid with the overall prediction given limited training data. In summary, we perform the following experiments.

- Disaggregated appliance electricity usage prediction using varying history size from the same user.
- Disaggregated appliance electricity usage prediction using the varying appliance usage history from 10 different users.
- Note that unless otherwise stated all experiments were performed with joint training, where electricity demand is predicted for all appliances jointly.

The symmetric mean absolute percentage error (SMAPE) is used for performance evaluation. SMAPE is defined as $SMAPE = \frac{100}{n}\sum\frac{|F_t - A_t|}{(F_t + A_t)/2}$ where A_t is the actual value and F_t is the forecast value. SMAPE was chosen because it is the standard time-series forecasting metric used in many benchmarks including the standard M3 benchmark [18] widely used to compare time-series models.

Results: The results in Fig. 2 indicate that the network is able to predict the disaggregated appliance usage significantly better when using the supporting information from the neighbors of the target household. Furthermore, one of the contribution of our paper is to forecast the disaggregated appliances jointly, instead of building a separate model for each appliance as was done in previous work. Figure 2 demonstrates a significant improvement in NILM accuracy by using joint learning. The intuition for this result is similar to the motivation of joint reconciliation during forecasting [19] which relies on the inherent correlation between the time-series in the same domain. During backward propagation the network learns the nonlinear interactions between the appliances and which improves the NILM performance. Note that the vanilla auto encoder per appliance (e.g., first category in Figure 2) is equivalent to the prior state of the art. A visual interpretation of the learned correlations amongst the appliances is part of our future work.

Note that only a relatively small number of extra users is required in order to realize a performance gain. Fig. 3 demonstrates the impact on performance as the number of available neighbors used for training is increased. There is a big drop as the second neighbor is added and then a gradual decrease in SMAPE is seen as additional households are used for target household NILM.

Our next experiment looks at performance impact as a function of the noise amount added to the signals. We generate white noise parametrized by mean and standard deviation. Fig. 4 shows SMAPE as a function of σ. We adjust only the input signal with additional noise and leave the output signals (e.g., disaggregated appliances) untouched. Our results indicate that using extra neighbor information does not help with a large amount of noise, however the system still performs reasonably well with about 1.5*σ white noise added to the system, which demonstrates the robustness of our approach.

So far, we have picked the neighbors uniformly randomly. In our next experiment we study how picking neighbors based on similarity impacts NILM performance. We use the *Pearson* correlation as a measure of similarity defined as $\rho = \frac{cov(T_1,T_2)}{\sigma_{T_1}\sigma_{T_2}}$, where T_1 and T_2 are the two time-series being compared. We compute the similarity based on the household aggregate demand. Fig. 5 depicts our results based on neighbor similarity. Our hypothesis was that the performance would improve when using more similar neighbors in the training data. According to our experiments, however, picking neighbors uniformly randomly worked best. Our intuition for this result is that having homogeneously similar neighbors in the training data does not provide enough new information and picking the most dissimilar neighbors does not have enough user similarity in the dataset, hence the uniformly random neighbors selection provide the best middle ground.

Our next experiment measures the per appliance SMAPE performance. Table I shows the results. What we find is that for larger appliances with a more regular usage patterns (e.g., AC, heater) neighbor information helps to improve appliance usage

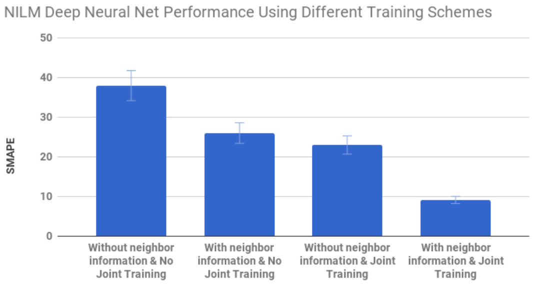

Fig. 2. Overall performance with and without joint training & using neighbor information for disaggregate appliance electricity usage prediction.

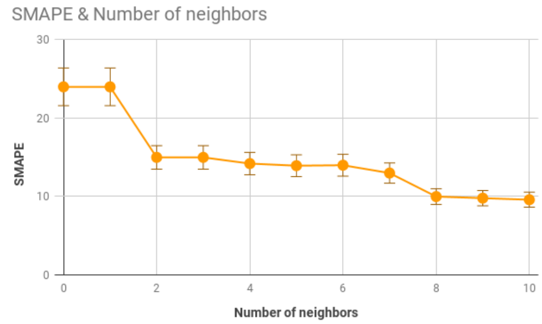

Fig. 3. Performance as the number of neighbors is increased.

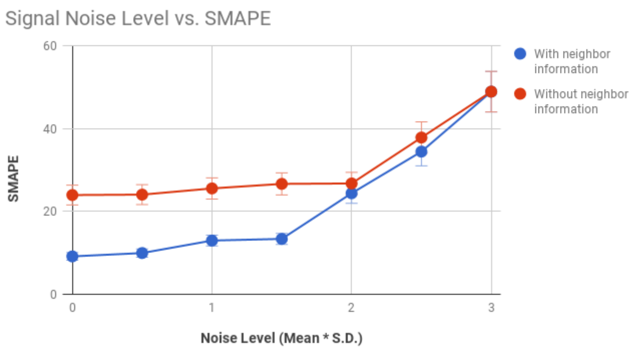

Fig. 4. Performance comparison as noise is added to the signal.

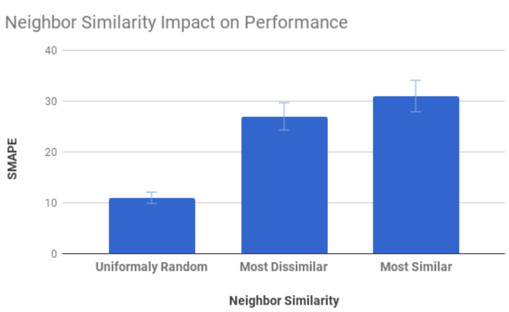

Fig. 5. Neighbors similarity impact on performance.

Appliance	With neighbor	Without neighbor
aggregate	10.10	21.10
air	15.71	38.67
aquarium	17.67	38.77
bathroom	19.48	41.00
bedroom	17.98	37.46
car	36.23	54.04
clotheswasher_dry	24.30	43.47
diningroom	22.51	39.26
dishwasher	58.36	71.73
disposal	44.01	58.77
freezer	32.72	48.11
furnace	24.77	43.19
garage	46.69	42.53
heater	20.76	37.99
housefan	20.11	38.05
icemaker	21.49	39.24
jacuzzi	23.91	40.79
kitchen	22.97	39.61
kitchenapp	50.23	56.59
lights_plugs	23.87	38.49
livingroom	25.05	41.60
microwave	35.10	50.30
office	25.41	41.07
outsidelights_plugs	25.43	40.59
oven	20.45	37.99
pool	19.96	37.46
poollight	20.78	37.71
pump	19.53	37.51
range	56.52	68.10
refrigerator	23.36	39.56
security	22.34	37.65
shed	24.41	40.08
sprinkler	20.36	37.67
utilityroom	31.56	44.57
venthood	27.60	42.71
waterheater	23.40	37.61

TABLE I
PER APPLIANCE NILM SMAPE PERFORMANCE (NO JOINT TRAINING).

disaggregation accuracy. For appliances with a less regular usage patterns (e.g., kitchen appliances) the performance is poor with and without additional neighbor information.

V. CONCLUSION

It is often the case that the forecasting performance suffers from short training size. This is especially a problem for NILM applications where disaggregating power consumption for a single household is challenging with a limited available history. We have demonstrated that by utilizing the neighboring information we are able to achieve a significantly better result. We have demonstrated the per-appliance performance impact under different neighbor selection criteria. The described approach is generally applicable to other domains where learning across domains is not only possible but has the potential of significantly improving forecasting accuracy.

In our future work we aim to study the diversity of time-series that leads to the best forecasting performance and explore the data complexity as a way for a better training set selection across the diverse training sources. A recurring theme is the unwillingness of time-series practitioners to use black-box models such as neural networks. Therefore, another direction for our future work is time-series neural network interpretability.

REFERENCES

[1] J. Kelly and W. Knottenbelt, "Does disaggregated electricity feedback reduce domestic electricity consumption? A systematic review of the literature," in *Proceedings of 3rd International NILM Workshop*, 2016.

[2] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computing*, vol. 9, no. 1, pp. 41–55, 1997.

[3] M. Assaad, R. Boné, and H. Cardot, "A new boosting algorithm for improved time-series forecasting with recurrent neural networks," *Information Fusion*, vol. 9, no. 1, pp. 41–55, 2008.

[4] N. Laptev, J. Yosinski, E. Li, and S. Smyl, "Time-series extreme event forecasting with neural networks at uber," *International Conference on Machine Learning*, 2017.

[5] G. W. Hart, "Nonintrusive appliance load monitoring," in *Proceedings of the IEEE*, vol. 80, no. 12, 1992, pp. 1870–1891.

[6] H. Kim, M. Marwah, M. F. Arlitt, G. Lyon, and J. Han, "Unsupervised disaggregation of low frequency power measurements," in *Proceedings of the 2011 SIAM International Conference on Data Mining*, 2011, pp. 747–758.

[7] J. G. Roos, I. E. Lane, E. C. Botha, and G. P. Hancke, "Using neural networks for non-intrusive monitoring of industrial electrical loads," in *Proceedings of Instrumentation and Measurement Technology Conference*, 1994, pp. 1115–1118.

[8] 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.

[9] N. Batra, A. Singh, and K. Whitehouse, "Neighbourhood nilm: A big-data approach to household energy disaggregation," *arXiv preprint arXiv:1511.02900*, 2015.

[10] S. M. Tabatabaei, S. Dick, and W. Xu, "Toward non-intrusive load monitoring via multi-label classification," *IEEE Transactions on Smart Grid*, vol. 8, no. 1, pp. 26–40, 2017.

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

[12] L. Jiang, J. Li, S. Luo, J. Jin, and S. West, "Literature review of power disaggregation," in *Proceedings of 2011 International Conference on Modelling, Identification and Control*, 2011, pp. 38–42.

[13] M. Zeifman and K. Roth, "Nonintrusive appliance load monitoring: Review and outlook," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 1, 2011.

[14] L. E. D. Christensen and B. Sparn, "Nilm applications for the energy-efficient home," National Renewable Energy Laboratory, Technical Report, 2012.

[15] T. SJ and L. B, "Forecasting at scale," *PeerJ Preprints 5:e3190v2*, 2017.

[16] C. Olah, "Understanding lstms," http://colah.github.io/posts/2015-08-Understanding-LSTMs/, 2015.

[17] Pecan Street Inc. (2017) Dataport from pecan street. [Online]. Available: https://dataport.cloud/

[18] R. Ulbricht, C. Hartmann, M. Hahmann, H. Donker, and W. Lehner, "Web-based benchmarks for forecasting systems: The ecast platform," in *Proceedings of the 2016 International Conference on Management of Data*, ser. SIGMOD '16. New York, NY, USA: ACM, 2016, pp. 2169–2172. [Online]. Available: http://doi.acm.org/10.1145/2882903.2899399

[19] R. J. Hyndman and G. Athanasopoulos, "Reconciling forecasts for hierarchical and grouped time series," 2014.