# WAVENILM: A CAUSAL NEURAL NETWORK FOR POWER DISAGGREGATION FROM THE COMPLEX POWER SIGNAL

*Alon Harell, Stephen Makonin, and Ivan V. Bajić*

School of Engineering Science, Simon Fraser University, Burnaby, Canada
Email: aharell@sfu.ca, smakonin@sfu.ca, ibajic@ensc.sfu.ca

## ABSTRACT

Non-intrusive load monitoring (NILM) helps meet energy conservation goals by estimating individual appliance power usage from a single aggregate measurement. Deep neural networks have become increasingly popular in attempting to solve NILM problems; however, many of them are not causal which is important for real-time application. We present a causal 1-D convolutional neural network inspired by WaveNet for NILM on low-frequency data. We also study using various components of the complex power signal for NILM, and demonstrate that using all four components available in a popular NILM dataset (current, active power, reactive power, and apparent power) we achieve faster convergence and higher performance than state-of-the-art results for the same dataset.

***Index Terms***— NILM, power disaggregation, convolutional neural network, causality

## 1. INTRODUCTION

As the cost and environmental impact of energy use continues to increase, the importance of power conservation and planning is growing significantly. Consumers and providers alike are encouraged to reduce power usage by many factors such as rising costs, legislation, and public image. Non-intrusive load monitoring (NILM), first proposed by Hart in 1992 [1], suggests disaggregating a single measurement of power from a central meter (e.g., a household smart meter) to estimate the power usage of different loads. Specifically, the aggregate power measurement $s(t)$ is given by

$$s(t) = \sum_{k=1}^{K} s_k(t) + n(t), \qquad (1)$$

where $s_k(t)$ is the power level of the $k$-th appliance and $n(t)$ is noise. The goal is to estimate $s_k(t), k = 1, 2, ..., K$, from $s(t)$, which is a form of *source separation*. This information can be used by the consumer to make informed decisions and by providers to educate clients and better plan production.

In the majority of NILM solutions, only one electrical measurement is considered, usually active/real power or current, and the output for each appliance is either a state classification (which is then related to a power level) or a direct regression of power used. We examine the effect of using multiple aggregate electric measurements (still collected by one central meter) to improve performance and convergence speed.

### 1.1. Active and reactive power

Power utility companies are generally concerned with two types of loads – active power ($P$) loads and reactive power ($Q$) loads. Active power loads (e.g., an electric stove) receive power from the grid and perform work dissipating it. Reactive power loads (e.g., a capacitor) receive power from the grid, store it, and after some period release it back to the grid in the opposite direction without dissipation. In many cases loads are both active and reactive – such as a heat pump or an air conditioning unit.

Mathematically, active power results from in-phase voltage and current, whereas reactive power results from out-of-phase voltage and current. Apparent power $S$, sometimes referred to as total momentary power, can also be a useful cue for disaggregation. These quantities are related as:

$$S = I \cdot V, \quad P = S \cdot \cos(\theta), \quad Q = S \cdot \sin(\theta), \quad (2)$$

where $I$ is current, $V$ is voltage, and $\theta$ is the phase of voltage relative to current (i.e., the phase angle).

### 1.2. Previous work

A number of recent papers on NILM present solutions based on deep neural networks. Kelly *et al.* first presented a deep learning-based approach in [2] and since then, a number of other attempts have been made at solving NILM using deep neural networks [3–5]. Although such solutions generally achieve good performance, other approaches still remain relevant in NILM. Some successful approaches include Hidden Markov Models (HMM) [6], as well as the cross entropy method [7], and integer programming [8, 9].
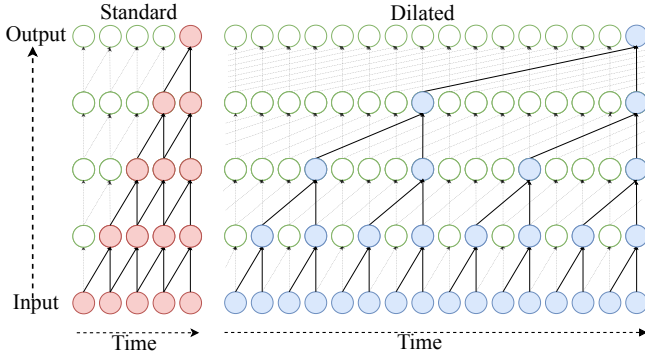
**Fig. 1**: Causal standard (left) and dilated (right) convolution stacks. Both have 4 layers, each with filter length 2. The dilation factor is increased by a factor of 2 with each layer. Colored nodes represent how output is calculated. Choices of colour simply differentiate one network from another and have no other significant meaning.

Among the recent NILM approaches using neural networks, convolutional networks [4] and bi-directional recurrent neural networks [2, 5] have been proposed. All these approaches are non-causal, requiring future data in order to disaggregate the current sample. In practical applications this means significant delay in disaggregation, effectively preventing real-time use. Our solution is causal, as well as computationally efficient, allowing for true on-line disaggregation.

Other solutions for NILM, such as HMMs, require modeling the problem in a certain manner, limiting the ability to easily adapt the solution to changes in the model or the data. We will demonstrate that performance can be greatly improved by using additional input measurements, a change that would require significant adaptation to [6] and other HMM based NILM solutions. The structure we propose, referred to as WaveNILM, requires no significant alterations whether using one or many input signals. This approach was recently explored in neural networks [4] and in mixed-integer linear programming [9], demonstrating improved performance when using reactive power along with the standard active power. WaveNILM builds on this, examining the benefits of additional input signals while achieving significantly better disaggregation performance than these earlier works and maintaining causality.

## 2. PROPOSED SOLUTION

### 2.1. Dilated causal convolutional neural networks

Maintaining causality is important in NILM as it allows for disaggregated data to be made available to users in real-time. This is especially true in a dynamically priced power grid, where a user might turn on a high energy appliance at a time where power is expensive. If given a real-time notification, the user may choose to defer the task to a later time, saving money and reducing load on the network at peak time [10].

Causal convolutional neural networks (CNNs) differ from

standard CNNs by using only samples from previous time-steps to calculate the current output. A stack of standard causal convolution layers requires long filters or very deep structures in order to achieve sufficiently large receptive fields. One way to address this problem is dilated causal convolution, as introduced in WaveNet [11]. In a dilated causal convolution with $x[n]$ as the input, dilation factor $M$ and length $N$ with parameters $c_k$, the output $y[n]$ is:

$$y[n] = \sum_{k=0}^{N-1} c_k \cdot x[n - M \cdot k]. \tag{3}$$

Note that receptive field of $y$ is of size $M \cdot (N - 1) + 1$, even though it only has $N$ parameters. By stacking dilated causal convolutions with increasing dilation factors we can achieve large receptive fields with a limited number of parameters while still maintaining causality, sampling rate, and using all available inputs. Fig. 1 provides a visualization of dilated convolution stacks.

### 2.2. Proposed network structure

The basic building block of WaveNILM is a gated version of the dilated causal convolutional layer, also inspired by WaveNet [11]. Samples from the current and past time steps are used as input to the dilated casual convolutions. Then, the output of each convolutional layer is used as an input to both a sigmoid activation (the relevance estimator or "gate") and a rectified linear activation (the regressor). The two activation values are then multiplied, and used as the output of the block. Each block output is then duplicated, one part being used as an input to the next layer, while the other (the "skip connection") skips all subsequent convolutions and is used in the final layers of WaveNILM. Each of these layers also contains a dropout of 10%.

In order to determine the appropriate architecture of WaveNILM, we first examined the size of the receptive field, which relates to the length of meaningful temporal relationships in the data. We found the appropriate length to be 512 samples, which is approximately 8.5 hours with 1-minute sampling. We then examined various possibilities for the number of layers and the layer size (number of filters) to arrive at the WaveNILM model architecture. In the final configuration (Fig. 2),the inputs are first fed into a 512-node time-distributed fully connected layer (with no connections between separate time samples), followed by a stack of 9 gated building blocks with 512, 256, 256, 128, 128, 256, 256, 256, 512 filters each. Skip connections from each layer of the stack are then concatenated, followed by another fully connected layer with $\tanh$ activation used as the output mask. Because WaveNILM uses multiple inputs, we multiply the mask only by the input relating to the quantity we wish to disaggregate. In total, WaveNILM contains approximately 3,250,000 trainable parameters, though this can vary slightly depending on input and output dimensionality.
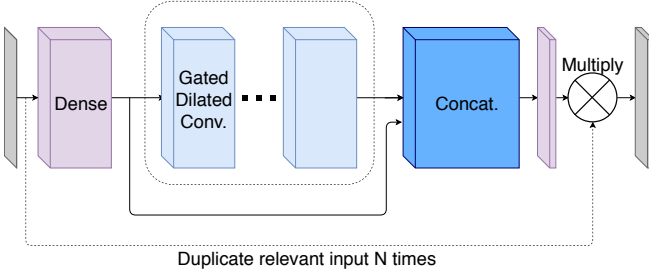
**Fig. 2**: WaveNILM network structure

## 3. EXPERIMENTAL SETUP

### 3.1. Data

There are many different datasets for NILM, e.g. [12–15] and more, with varying properties such as sampling frequency, duration, location, etc. We chose to focus on AMPds2 [13], a low frequency dataset (1 minute sampling), taken from one house in Canada over two years. The measurement includes the aggregate signals and specific data from 20 sub-meters. Each measurement contains current, apparent power, active power, and reactive power. AMPds2 is an extension of a previous dataset known as AMPds [16], which contained the same measurements, but only for the first year.

Deferrable loads were defined by [6] as large loads that the user could choose to run at lower levels or at off-peak hours. Successfully disaggregating these appliances is of significant importance to both users, in reducing power cost, and to suppliers, in helping to prevent brownouts by reducing peak time power consumption (known as peak shaving [17]). In AMPds2 these appliances are the HVAC system, the heat pump, the wall oven, the clothes dryer, and the dishwasher. When disaggregating only a subset of all loads, the aggregate signal contains many signals from other loads, as well as the desired signals. We consider these signals from other loads to be the measurement noise (with respect to the desired loads) and consider both noisy and denoised cases. In the noisy case, the inputs are actual aggregate measurements wheareas in the denoised case the inputs are the sum of all target appliance measurements (ground truth signals). On average, a given aggregate reading in AMPds contains about 60% noise when trying to disaggregate deferrable loads [6].

Other work using AMPds or AMPds2 uses only denoised scenarios [9] or achieves significantly inferior results [4]. For these reasons, we consider [6], which is based on a sparse super-state HMM (SSHMM), to be the current state-of-the-art for AMPds and use it for comparison with WaveNILM. At the time of its publication, AMPds2 had not yet been available, meaning [6] was based on AMPds. To ensure a fair comparison with [6], WaveNILM was trained and tested on the same data and scenarios as [6], and the results on the full AMPds2 dataset are reported separately.

We suggest four possible inputs, as collected in AMPds2: current ($I$), active/real power ($P$), reactive power ($Q$), and apparent power ($S$). For each scenario we compare the per-

formance of each input used individually, as well as using a combination of two inputs ($P$ and $Q$), and finally using all four inputs. Household power cost is based, in general, only on active power $P$. For this reason, whenever $P$ was available in the scenario, it was used as the output unless otherwise needed for comparison to previous work. In order to explore the effect of using multiple inputs while still maintaining comparability to previous work, we run various combinations of the parameters of each experiment.

### 3.2. Training and testing

Training was performed on one day samples (1440 time-steps) with a batch size of 50 samples. Data was normalized by the next power of 2 to the maximum of the aggregate data. This can be thought of as scaling the entire meter range to $[0, 1]$. Because the receptive field of WaveNILM is 512 samples, disaggregating the present measurement requires the 511 previous samples. To avoid training the network on incomplete data, loss was computed for samples 512-1440. This required creating overlap in the training samples so that all available data was used for both training and testing.

Training was conducted using 90% of the data and testing on the remaining 10%. Full disaggregation and deferrable load scenarios were trained for as many as 500 and 300 epochs, respectively, allowing for longer training on the more complex problem.

### 3.3. Metrics

Because waveNILM is a regression network, we focus on the error in disaggregated power, as opposed to the state of the appliances. A common metric for evaluating disaggregated power, is Estimated Accuracy, defined by Johnson and Willsky [18] and is one of the well accepted accuracy metrics [19]:

$$Est.Acc. = 1 - \frac{\sum\limits_{t=1}^{T} \sum\limits_{k=1}^{K} |\widehat{s}_k(t) - s_k(t)|}{2 \sum\limits_{t=1}^{T} \sum\limits_{k=1}^{K} s_k(t)}, \qquad (4)$$

where $\widehat{s}_k(t)$ is the predicted power level of appliance $k$ at time $t$, $s_k(t)$ is the ground truth, and $T$ and $K$ are the total time and the number of appliances, respectively. The above expression yields total estimated accuracy; if needed, the summation over $k$ can be removed creating an appliance-specific estimation accuracy [19].

## 4. RESULTS

### 4.1. Different input signals

We studied the performance of WaveNILM with a variety of input signals. When using only one input, the same electrical quantity was used as the output. When using several inputs,
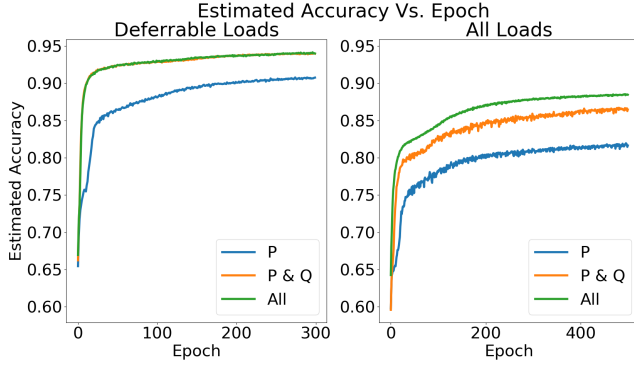
**Fig. 3**: Convergence speed for single vs. multiple inputs. Note that the on the deferrable loads case (left), the training with P & Q and training with all inputs produce almost identical curves.

we choose active power $P$ or current $I$ as the output. In order to study the significance of input and output signals we compare partial-disaggregation on the deferrable loads with real aggregate input (noisy case), as well as the full 20-load disaggregation.

Examination of the results, as summarized in Table 1, shows the significant improvement in performance with the use of multiple inputs. In addition to final performance, convergence time was also greatly improved when using additional inputs, as seen in Fig. 3. In both aspects, the main benefit is achieved when introducing reactive power $Q$ as an input, with subsequent additions providing marginal improvement. Note that $Q$ as a sole input provides excellent results, however since domestic consumers are not charged for it, it's disaggregation alone does not help achieve the goals of NILM as explained in Section 1.

**Table 1**: Noisy case results with various inputs, AMPds2

| Input Signal | All loads | Deferrable loads |
|:---:|:---:|:---:|
| $I$ | 85.6% | 92.0% |
| $P$ | 82.6% | 90.9% |
| $Q$ | **91.1%** | 94.4% |
| $S$ | 86.7% | 88.9% |
| $P$ and $Q$ | 87.5% | 93.9% |
| All 4 (output: $P$) | 88.4% | 94.2% |
| All 4 (output: $I$) | **90.2%** | **95.0%** |

### 4.2. Comparison with SSHMM

When comparing with SSHMM [6], only the first year of AMPds2 (i.e., AMPds) was used and the results were all 10-fold cross-validated meaning we repeated each experiment ten times, changing the test set for each iteration, and finally, average the performance over all ten tests. Both noisy and denoised cases are examined, as explained in Section 3.1. For each case, the output is the current $I$, as in [6]. The input is either $I$, as in [6], or all four signals $I, P, Q, S$. Denoised case results are shown in Table 2, with the best result indicated in bold. As seen in the table, the performance on denoised

aggregate signals was near perfect.We consider this problem largely solved.

**Table 2**: Denoised case results on deferrable loads, AMPds

| Disaggregator | Input | Output | Est. Acc. |
|:---:|:---:|:---:|:---:|
| WaveNILM | $I$ | $I$ | 99.0% |
| WaveNILM | $I, P, Q, S$ | $I$ | **99.1%** |
| SSHMM [6] | $I$ | $I$ | 99.0% |

Next we consider the noisy case, where the actual aggregate measurement was used as an input. Again we compare the same configurations of WaveNILM as in the denoised case, and all experiments were 10-fold cross-validated. As seen in Table 3, when compared with SSHMM [6], WaveNILM performs slightly worse when using only $I$ as an input, and slightly better when using all four inputs. Run times are also improved, with WaveNILM completing disaggregation of one sample in under $150\mu s$ for all configurations, where the SSHMM required approximately $4.55ms$ (though both are appropriate for real-time disaggregation). More importantly, the run time (and storage space) of WaveNILM increases only marginally when increasing the number of loads, the difference between the simplest configuration (1 input, 5 loads) and most complicated (4 inputs, all 20 loads) being under $20\mu s$.

**Table 3**: Noisy case results on deferrable loads, AMPds

| Disaggregator | Input | Output | Est. Acc. |
|:---:|:---:|:---:|:---:|
| WaveNILM | $I$ | $I$ | 92.3% |
| WaveNILM | $I, P, Q, S$ | $I$ | **94.7%** |
| SSHMM [6] | $I$ | $I$ | 94.0 % |

## 5. CONCLUSIONS

We presented WaveNILM, a flexible and causal CNN for load disaggregation. One of its greatest advantages over existing NILM solutions is the ability to easily add various inputs to help improve disaggregation. WaveNILM requires only 512 new parameters for each new input signal, and only 3072 parameters for each added output (load). This is particularly important when exploring additional external measurements that have been shown to be useful for disaggregation such as weather data [20], or time of day [21].

Reactive power has been shown to be of particular benefit to NILM, providing most of the performance benefit, with apparent power and current only adding marginal improvement. As far as data collection is concerned, we believe the results presented here should encourage the recording of as many electrical signals as possible, with active and reactive power being the most significant. Reactive power is not currently reported by most smart meters, yet it can be calculated within the meter. This means that reactive power is, in principle, an available measurement, and could be more accessible in future generations of smart meters.

# 6. REFERENCES

[1] G. W. Hart, "Nonintrusive appliance load monitoring," *Proc. IEEE*, vol. 80, no. 12, pp. 1870–1891, Dec. 1992.

[2] J. Kelly and W. J. Knottenbelt, "Neural NILM: Deep Neural Networks Applied to Energy Disaggregation," *CoRR*, vol. abs/1507.06594, 2015.

[3] J. Kim, T. T. H. Le, and H. Kim, "Nonintrusive Load Monitoring Based on Advanced Deep Learning and Novel Signature," *Computational Intelligence and Neuroscience*, vol. 2017, Article ID 4216281, 22 pages, 2017. https://doi.org/10.1155/2017/4216281.

[4] M. Valenti, R. Bonfigli, E. Principi, and a. S. Squartini, "Exploiting the Reactive Power in Deep Neural Models for Non-Intrusive Load Monitoring," in *2018 International Joint Conference on Neural Networks (IJCNN)*, July 2018, pp. 1–8.

[5] H. Rafiq, H. Zhang, H. Li, and M. K. Ochani, "Regularized LSTM Based Deep Learning Model: First Step towards Real-Time Non-Intrusive Load Monitoring," in *2018 IEEE International Conference on Smart Energy Grid Engineering (SEGE)*, Aug. 2018, pp. 234–239.

[6] S. Makonin, F. Popowich, I. V. Bajić, B. Gill, and L. Bartram, "Exploiting HMM sparsity to perform online real-time nonintrusive load monitoring," *IEEE Transactions on Smart Grid*, vol. 7, no. 6, pp. 2575–2585, 2016.

[7] R. Machlev, Y. Levron, and Y. Beck, "Modified Cross-Entropy Method for Classification of Events in NILM Systems," *IEEE Transactions on Smart Grid*, pp. 1–1, 2018.

[8] M. Z. A. Bhotto, S. Makonin, and I. V. Bajić, "Load disaggregation based on aided linear integer programming," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 7, pp. 792–796, July 2017.

[9] F. M. Wittmann, J. C. Lopez, and M. J. Rider, "Nonintrusive load monitoring algorithm using mixed-integer linear programming," *IEEE Transactions on Consumer Electronics*, vol. 64, no. 2, pp. 180–187, May 2018.

[10] K. Ehrhardt-Martinez, K. A. Donnelly, S. Laitner, et al., "Advanced metering initiatives and residential feedback programs: a meta-review for household electricity-saving opportunities," American Council for an Energy-Efficient Economy Washington, DC, 2010.

[11] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[12] 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, pp. 150007, 2015.

[13] S. Makonin, B. Ellert, I. V. Bajić, and F. Popowich, "Electricity, water, and natural gas consumption of a residential house in Canada from 2012 to 2014," *Scientific Data*, vol. 3, no. 160037, pp. 1–12, 2016.

[14] J. Z. Kolter and M. J. Johnson, "REDD: A public data set for energy disaggregation research," in *Workshop on Data Mining Applications in Sustainability (SIGKDD), San Diego, CA*, 2011, vol. 25, pp. 59–62.

[15] T. Picon, M. Nait Meziane, P. Ravier, G. Lamarque, C. Novello, J.-C. Le Bunetel, and Y. Raingeaud, "COOLL: Controlled on/off loads library, a public dataset of high-sampled electrical signals for appliance identification," *arXiv preprint arXiv:1611.05803 [cs.OH]*, 2016.

[16] S. Makonin, F. Popowich, L. Bartram, B. Gill, and I. V. Bajić, "AMPds: A public dataset for load disaggregation and eco-feedback research," in *2013 IEEE Electrical Power Energy Conference*, Aug 2013, pp. 1–6.

[17] B. Roberts, "Shaving load peaks from the substation," *POWER Magazine*, Oct. 2006.

[18] M. J Johnson and A. S. Willsky, "Bayesian nonparametric hidden semi-Markov models," *Journal of Machine Learning Research*, vol. 14, no. Feb., pp. 673–701, 2013.

[19] S. Makonin and F. Popowich, "Nonintrusive load monitoring (NILM) performance evaluation," *Energy Efficiency*, vol. 8, no. 4, pp. 809–814, 2015.

[20] A. Harell, S. Makonin, and I. V. Bajić, "A recurrent neural network for multisensory non-intrusive load monitoring on a Raspberry Pi," in *IEEE MMSP' 18*, Vancouver, BC, Aug. 2018, demo paper, electronic proceedings.

[21] C. Dinesh, S. Makonin, and I. V. Bajić, "Incorporating time-of-day usage patterns into non-intrusive load monitoring," in *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Nov. 2017, pp. 1110–1114.