Informatics Research Proposal:
"Can a neural network summarize your electricity use?"

Christopher Sipola (s1667278), suprvised by Nigel Goddard

April 2017

Abstract

This is a research proposal for a MSc in Artificial Intelligence dissertation project at the University of Edinburgh. The project is part of the larger IDEAL project, which explores whether personalized feedback on energy usage can help homeowners reduce their energy bills. The goal of the dissertation project is to determine whether a neural network is capable of summarizing the electricity usage of three appliances—the refrigerator, kettle and washing machine—given only the aggregate electricity signal of a smart meter. In addition to informing homeowners of their usage, the summary statistics can also be used as inputs to models that aim to disaggregate the main signal into those of the individual appliances.

1 Introduction

Providing personalized feedback to homeowners on electricity usage can result in lower energy consumption, with some research putting the cost savings at 5–12% of the total energy bill. It has been shown that the most successful feedback is given frequently (e.g., daily), is tied to behavior and includes a breakdown by appliance [1]. Example feedback could be: "Your washing machine ran four times in the last week, using 6 kWh and costing 90p. Over a year, this translates into about 300 kWh and £45." These sort of energy savings are the goal of the IDEAL research project at the University of Edinburgh, which explores the use of smart home technology for generating personalized feedback [2].

Although smart meters typically report household electricity usage data every few seconds, they do not provide a breakdown by appliance. Non-intrusive load monitoring (NILM), or energy disaggregation, is a set of techniques that aim to break the aggregate signal into the signals of the appliances [3]. Since NILM is a supervised learning task, disaggregation models must be trained using appliance-level data. There are a number of datasets containing appliance-level data, often of varying sample rates, number of homes and study period lengths. Arguably one of the best datasets for NILM is REFIT, which contains aggregate- and appliance-level electricity usage data for 20 UK homes at a high sample rate and a study period of nearly two years (see Appendix) [4].

To calculate appliance-level summary statistics, it is possible to use NILM models to predict the appliance signals and then summarize the resulting predictions. However, building a model

that predicts the summary statistics *directly* may greatly simplify the process, since calculating point-by-point appliance signals is expensive.

The goal of the proposed dissertation project is to support IDEAL by exploring whether a neural network can summarize the energy usage statistics of individual appliances given only the aggregate signal. The REFIT dataset will provide more diversity across appliance models and consumption behaviors than previous datasets—which is important for data-hungry neural network models if they are to generalize to unseen time periods and unseen homes. Other researchers have explored the use of neural network models in NILM, but not for the task of summarizing energy usage [5, 6].

There are two main uses of a model that can predict appliance-level energy statistics. One is to present the information in a way that is informative and actionable for homeowners who would like to change their energy consumption behavior. The other is to use the summary statistics as inputs for other NILM models. One such model is an additive factorial hidden Markov model (AFHMM) where latent Bayesian melding (LBM) is used to incorporate appliance-level statistics for more accurate predictions [7]. The AFHMM with LBM is already state-of-the-art in NILM, and it may see further performance gains if the appliance-level statistics were calculated for each home instead of using nationally representative averages, since appliance models and usage behaviors can vary greatly between households.

2 Background

The modeling task of this project is unique because there is no major research on using neural networks for regression on time series signatures. Additionally, the input signal is very "noisy" in that it is an aggregation of signals when we only care about one. For this reason, this project will have to combine ideas from several different bodies of research, including image recognition, time series modeling and NILM.

In recent years, there have been great advances in the field of image recognition [8]. This success is largely due to the revival of the neural network—and specifically, the convolutional neural network (CNN). A neural network is a versatile and powerful machine learning model that is loosely based on the structure of a biological brain. It is modeled as neurons or "units" that are connected as an acyclic graph (figure 1). Layers are comprised of units, and units of one layer can be combined to create the units of other layers. The power of the model comes from the nonlinear functions[1] that are applied after the linear combinations, allowing the network to model interactions between features and represent increasingly abstract relationships in the data. Neural networks are now state-of-the-art in image recognition, machine translation, speech recognition, text prediction, and other fields in machine learning.

A CNN is a type of neural network that exploits strong local correlations between input features and is therefore specially suited for dealing with image data. It works by sliding small feature detectors[2] over the image, identifying pixel configurations such as edges and curves. Units created with this process can then be fed into feature detectors that identify more abstract features. With enough convolutions, the network can learn complex representations such as human faces. The weights of a feature detector are shared between receptive fields, greatly reducing the number of parameters to be learned by the model [10].

[1] Common nonlinear functions are the rectified linear unit (ReLU) function $f(x) = \max(0, x)$ and the sigmoid function $f(x) = 1/(1 + e^{-t})$.

[2] These are also known as a kernels.

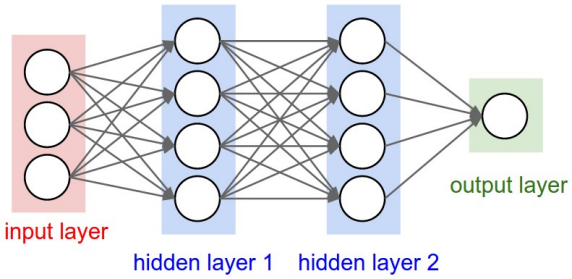

Figure 1: Basic neural network architecture with two hidden layers, from [9].

Time series data often have strong correlations between timesteps and can therefore be thought of as a one-dimensional image, which allows researchers to adapt convolutional neural networks for time series classification.[3] Instead of an edge, the feature detectors may identify an increase in the time series values; and instead of a human face, they may detect the signature of a heartbeat. The state-of-the-art for time series classification has been k-nearest neighbor classification (k-NN) combined with dynamic time warping (DTW), which works by "warping" time to line up differently-spaced time series and then selecting the labeled time series with the most similar pattern (usually *1*-NN) [11, 12]. Another method for times series classification is the multi-channel deep CNN, which takes multiple time series as separate "channel" inputs to a CNN in the same way that it would take a color image as three separate channels (typically red, green and blue) [13]. Multi-channel CNNs have been shown to be more efficient while having competitive accuracy when compared to k-NN with DTW.

When applied to high-dimensional inputs like audio data or energy signal data, CNNs can benefit from dilated convolutions, which use large but sparse feature detectors that "increase the receptive field by orders of magnitude, without greatly increasing computational cost" [14]. There have been efforts to represent time series as two-dimensional objects in order to exploit techniques from image recognition more directly [15], but these exotic representations are not widely used [16].

The most effective appliance signal prediction neural network model is PointNet, which uses a CNN to predict the signal of one appliance at one timestep given a window of the aggregate signal. PointNet was tested on five appliances: the kettle, microwave, fridge, dish washer, and washing machine. Each appliance had its own architecture with an input window length that was adjusted for the amount of time each appliance is typically used [5].

3 Methods

3.1 Overview of task

The project will predict the daily energy used by three appliances:

1. the **refrigerator**, which has a regular, frequent and distinctive signature (figure 2);

[3] Although a model called a recurrent neural network is often used for forecasting sequential data—such as financial and language data—our task is a classification problem and is therefore more likely to benefit from image recognition techniques.

2. the **kettle**, with its simple rectangular signature, lending it to relatively easy disaggregation by neural network models [5]; and

3. the **washing machine**, which has a very distinctive but complex signature that can last for hours [4].

The three signatures were chosen because they provide a good variety of signatures and usage patterns for testing the model. The appliances are also used frequently enough that we expect there will be enough data for the neural network. Each appliance will have its own architecture based on its time window of a typical use.

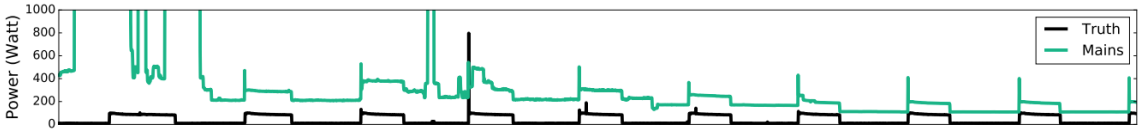

Figure 2: Example fridge signal ("Truth") and aggregate signal ("Mains"), taken from [5].

3.2 Data preparation

The data will be read and stored in a SQLite database. The main SQLite table containing the data will have "melted" data with columns for datetime, home, signal type (either aggregate or that of a specific appliance), and the associated signal. This format will allow for easy sampling of the data. Any arithmetic to subtract or add signals will be done in the a preprocessing step by the model.

Although the "cleaned" version of the dataset will be used, there will likely have to be some additional cleaning and data exploration to ensure that the data is in line with expectations and is suitable for modeling. For example, periods with power outages or other irregularities will likely be ignored.

3.3 Data provider system

The data provider will query the SQLite data base, preprocess the data and perform data augmentation. This process likely be more intensive relative to other projects, for three main reasons: (1) determining when an appliance is active even when given its signal is nontrivial; (2) data augmentation may itself involve its own form of sampling; and (3) subtracting appliance signals from the aggregate may leave residual spikes since signals are not completely aligned in time, perhaps eliminating subtraction as a possible operation when performing data augmentation.[4]

The data provider system needs to be able to select a window of data where the target appliance is being used so that positive and negative training samples are at least somewhat balanced. The system will sample at 50:50 for the positive-negative ratio, which is what is used by [6]. In order to sample, rules must be created to determine whether an appliance is being used given the appliance's signal and some heuristics about its typical usage duration, maximum power, on-power threshold,

[4]For example, a preliminary exploration of one home's data showed that the aggregate signal was ∼12 seconds behind of one or many of the individual appliances.

average power while in use, and standard deviation on power [5]. It may be possible to adapt some code and methods from the open-source NILMTK package[5] or other NILM research [6, 5].

The window size will be set to one full day, since it is natural to calculate summary statistics at the daily level. However, smaller windows will also be tested since "increasing window size hurts disaggregation performance for short-duration appliances" [6].[6] To be sure the model does not overfit for specific household routines that are strongly correlated with time of day, the windows will start at random datetimes.[7]

3.4 Data preprocessing

In addition to typical data standardization techniques, the project will test whether first-differencing the training signal improves learning.[8] This transformation would make it so that a given appliance signature in the aggregate signal is more or less unaffected by the signals of the other appliances. Although a neural network is, in principle, capable of learning first-differencing, it may make sense to exploit this information if it is known beforehand to be a better representation of the data. Also, first-differencing has the added benefit of naturally standardizing the data so that it has a mean of zero, effectively ignoring vampire loads—that is, loads that are always on.

3.5 Data augmentation

Neural networks require a lot of data since there are so many trainable parameters. To increase the amount of data, researchers commonly use data augmentation, or the expansion of the training set with realistic distortions to the input data [21, 22]. This is made easier in NILM since we can create synthetic aggregate signals by adding appliance signals. NILM researchers have done just this, using a 50:50 real-to-synthetic ratio, where synthetic data has a 50% chance of containing the target appliance and a 25% chance of containing each other appliance [6].

Since this naive method ignores cross-appliance usage patterns, this project will also try techniques to preserve some structure within the real aggregate data. One such method is randomly swapping each appliance signal for that of another home (see Appendix, figure 5).

3.6 Model architecture

The project will use a model similar to the convolutional neural network architecture in [5] as a starting point for pilot tests since it is the most successful type of neural network applied to NILM (figure 3). However, there will have to be an adjustment to the window size ("Window length" in figure 3), since the input window for this project will be for a full day. It may make sense to compress the data before some of the more intensive computation in the network given that a full day of data has over 14,000 dimensions. Therefore, pilot tests will also test whether the heavy use

[5] In particular, the NILMTK method `Electric.get_activations()` could provide guidance.

[6] Using window sizes that are too small will require a methodology for accounting for signatures that are split across windows. Using overlapping windows may seem like an intuitive solution, but a methodology must then be created to account for duplicate energy predictions. Of course, if using a full day as the window size, this problem still exists since signatures can still be split by the edges of the window—but the overlap would be less.

[7] See section 6 for an extension that allows the model to learn when time of day can be important for predicting total energy usage.

[8] To first-difference a series $\mathbf{y} = \{y_1, y_2, ..., y_T\}$ is to subtract off its lag $L\mathbf{y} = \{y_0, y_1, ..., y_{T-1}\}$ to create the differenced series $\Delta\mathbf{y} = \mathbf{y} - L\mathbf{y} = \{y_1 - y_0, y_2 - y_1, ..., y_T - y_{T-1}\}$, where the lag operator L is defined as $L^k y_t = y_{t-k}$ and the differencing operator is defined as $\Delta y = (1 - L)y$.

of pooling layers to reduce the size of the data, or an initial low-dimensional fully connected layer, is more effective. Instead of—or in addition to—reducing the size of the data, dilated convolutions can also be tested.

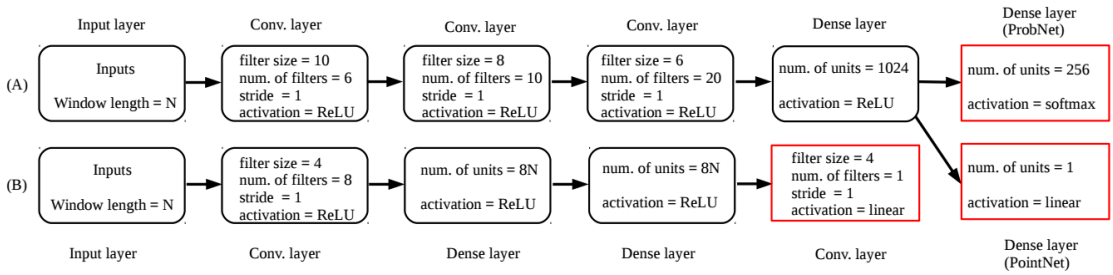

Figure 3: Architectures described in [5]. The architecture of interest for this project will be (A), following the arrows to the PointNet dense layer.

4 Evaluation and outcomes

Before training, the dataset will be split into a training set and a test set. The split will be made so that the test set has data for *both* (1) unseen time periods for homes for which we trained the model; and (2) unseen homes. This will test the model's ability to generalize on both unseen days and unseen homes. One way to do this is to select some number of homes—say, 16—and use 80% of their data for training. The other 20% of the data for the 16 homes, as well as *all* the data for the other 4 homes, could be used for testing. The homes will have to be carefully selected to ensure that they contain data for the necessary appliances.

The main performance metric of interest will be the mean absolute percentage error (MAPE) for each appliance, averaged across days for which we have test data. Since there is no similar research on energy summarization, there are no standards to which the performance can be directly compared. However, a qualitative and intuitive evaluation can still be made of the usefulness of the data to homeowners. For example, if the MAPE is above 50%, then this is unlikely to inspire trust and lead to behavioral changes. But even if the error is large, the data can still be useful as an input to other NILM models if it is more accurate than nationally representative averages.[9]

The outputs of the analysis will be summary statistics that can be used to help homeowners reduce their energy bills. The data can also be used as an input to other NILM models, such as the AFHMM with LBM (section 1).

5 Plan

The plan is described below. Figure 4 shows the project timeline.

1. **Literature review (9 days).** This includes research on time series classification and modeling (especially using neural networks), regression using neural networks, the REFIT data source and related work in NILM.

[9] See section 6 for extensions on using PointNet or other signal disaggregation methods for comparison.

2. **Import, process and clean data (5 days).**

3. **Explore data (3 days).** Get a sense of the shape and diversity of the signals of the three appliances, as well as the aggregate signal. Plot some example signals for the report.

4. **Build data provider system (14 days)** that can effectively sample the data and perform data augmentation. This will require getting familiar with Keras, building a prototype model and checking for reasonable results.

5. **Build modeling system (16 days).** Conduct pilot tests to see how the prototype model reacts to changes in architecture and hyperparameters. Design the architectures of the models that will be tested. Build system for automatic model testing. Run system.

6. **Evaluate results (4 days).**

7. **Re-run parts of analysis if necessary (10 days).**

8. **Write report draft (ongoing, 60+ days).** This starts with the literature review and will be concurrent with the programming and analysis.

9. **Finalize report (14 days).**

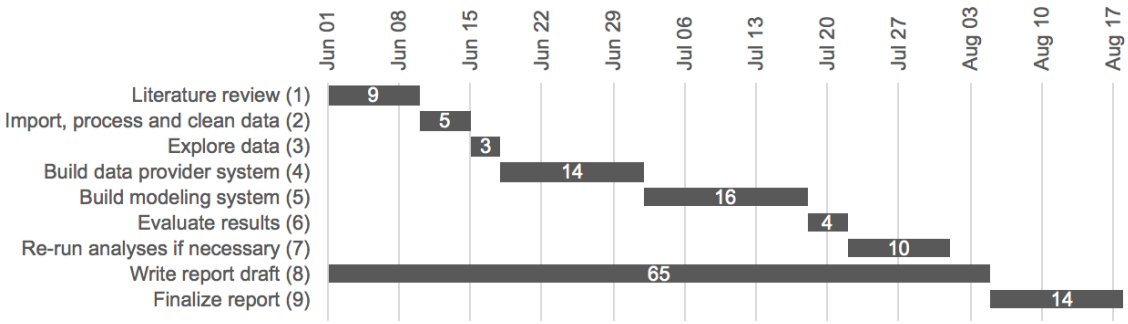

Figure 4: The project timeline with bars indicating the days over which task should be completed. The bars are labeled with the duration of the task in days. The exact dates are approximate; in reality they will likely bleed into one another.

6 Extensions

Possible extensions are below, with some slight preference for the those listed first due to high estimated reward when compared to time cost. However, these preferences may change over the course of the project.

- **Evaluating an additional target summary variable**, namely, the number of times each appliance is used. This may require additional research on counting using neural networks [23, 24]. One initial concern with this extension is that most research applications have been applied to images where there are many instances of the objects to be counted, allowing for

false positives and false negatives to effectively cancel each other out. An appliance like a kettle is not used so often for this effect to take hold.

- **Capturing time-of-day information in the neural network** to account for typical temporal usage patterns of appliances. If randomizing the start time of the input windows, then the CNN is blind to time of day due to the shared weights of the kernels. The theory for this extension is not well-explored, but one possible implementation could be to add timestamp information as additional channels so that the network can learn the logical operations to identify the time of day. The channels could be periodic functions or radial basis functions.[10]
- **Building architectures for additional appliances**, such as the dish washer or the microwave.
- **Evaluate how much the model improves the AFHMM LBM disaggregation method.**
- **Compare model output to that of PointNet or other signal disaggregation models on the REFIT data** by summing over the disaggregated signals produced by those models. While it is entirely possible that our model will be more computationally efficient, it is optimistic to think that it will have lower error.

Appendix

Data

There are a number of popular datasets with smart meter data, such as BLUED [17], HES [18], REDD [19], and UK-DALE [20]. REFIT (Personalised Retrofit Decision Support Tools for UK Homes Using Smart Home Technology) is the only data set that has the combination of a long study period (21 months), more than a handful of participating homes (20), a large number of appliances (9), and a high resolution (8 seconds).[11] In comparison, BLUED and HES have just 1 home and 4 homes (respectively), HES just 2 minute resolution and REDD a study length of just 3–19 days. The REFIT dataset was collected as part of the REFIT project, which is an interdisciplinary research endeavor with goals similar to those of the IDEAL project.

The data is in CSV format, with the data for each home stored in a separate file. On average, there are roughly 7–8 million observations for each of the 20 homes. The columns in each file are for the datetime, aggregate signal, and the appliance signals for up to nine appliances (with each appliance having its own column).

Pseudo-synthetic data augmentation

The proposed data augmentation process is described in figure 5.

[10] Since neural networks can learn logical operations such as AND and OR, in principle it is capable of learning that (for example) a spike in channel 1 (the aggregate signal) AND a large value for channel 2 (some periodic function) AND a small value for channel 3 (another periodic function) means that it is likely a kettle has been used. Adding these additional channels is likely best done in a hidden layer, since adding them to the input might increase computation time significantly.

[11] The standard deviation of the sequential difference between datetimes for the first few homes was usually above 6.0, which is quite a bit of variation. This only includes datetime differences under 30 seconds since larger values were rare—often accounting for less than 0.1% of the observations—and were likely due to power outages or other idiosyncrasies in the data. The mean of the differences were around 6 seconds and the median was consistently 7.

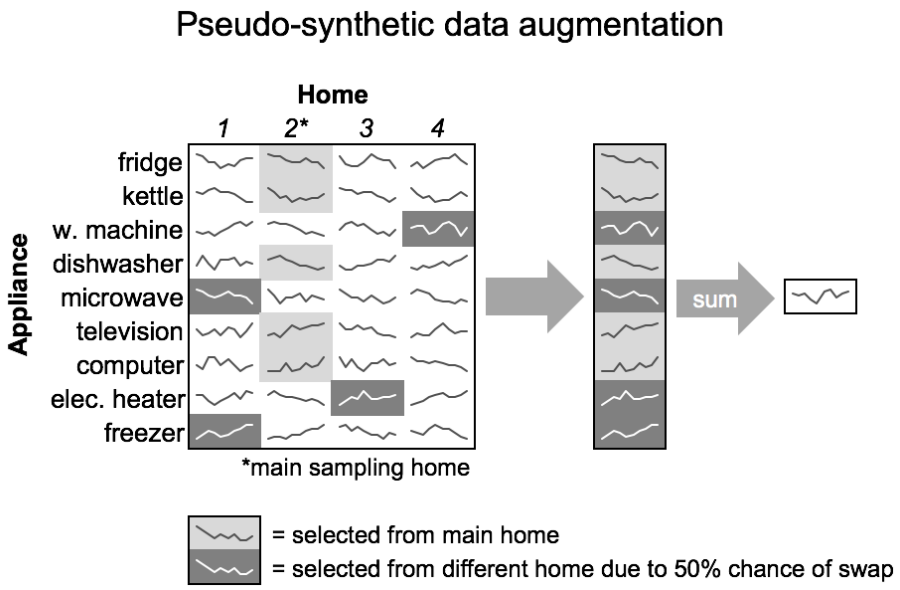

Figure 5: Assuming the dataset has just four homes and all homes have the same appliances, the process is as follows: (1) select a random time window and a random home (in the figure, home 2 has been selected); (2) select one appliance within that home (fridge); (3) with 50% probability, swap the signal of that appliance with that of another home over the same time period (no swap for the fridge); (3) repeat steps 2 and 3 for all other appliances (resulting in swaps for the washing machine, microwave, electric heater, and freezer); and (4) aggregate the selected appliance signals to create a pseudo-synthetic aggregate signal. Since the aggregate signal does not just come from the main signal,

References

[1] C. Fischer, "Feedback on household electricity consumption: a tool for saving energy?," *Energy efficiency*, vol. 1, no. 1, pp. 79–104, 2008.

[2] "IDEAL home energy advice project." Website available at http://www.energyoracle.org/. Accessed 04 Apr 2017.

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

[4] D. Murray, J. Liao, L. Stankovic, V. Stankovic, R. Hauxwell-Baldwin, C. Wilson, M. Coleman, T. Kane, and S. Firth, "A data management platform for personalised real-time energy feedback," *EU Science Hub*, 2015.

[5] C. Zhang, M. Zhong, Z. Wang, N. Goddard, and C. Sutton, "Sequence-to-point learning with neural networks for nonintrusive load monitoring," *arXiv preprint arXiv:1612.09106*, 2016.

[6] 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*, pp. 55–64, ACM, 2015.

[7] M. Zhong, N. Goddard, and C. Sutton, "Latent bayesian melding for integrating individual and population models," in *Advances in Neural Information Processing Systems*, pp. 3618–3626, 2015.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[9] J. J. Fei-Fei Li, Andrej Karpathy, "Cs231n: Convolutional neural networks for visual recognition." Stanford University course. Website available at http://cs231n.stanford.edu/. Accessed 13 Mar 2016.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[11] G. E. Batista, X. Wang, and E. J. Keogh, "A complexity-invariant distance measure for time series," in *Proceedings of the 2011 SIAM International Conference on Data Mining*, pp. 699–710, SIAM, 2011.

[12] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series data: experimental comparison of representations and distance measures," *Proceedings of the VLDB Endowment*, vol. 1, no. 2, pp. 1542–1552, 2008.

[13] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, "Time series classification using multi-channels deep convolutional neural networks," in *International Conference on Web-Age Information Management*, pp. 298–310, Springer, 2014.

[14] 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," *CoRR abs/1609.03499*, 2016.

[15] Z. Wang and T. Oates, "Encoding time series as images for visual inspection and classification using tiled convolutional neural networks," in *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[16] J. C. B. Gamboa, "Deep learning for time-series analysis," *arXiv preprint arXiv:1701.01887*, 2017.

[17] A. Filip, "Blued: A fully labeled public dataset for event-based non-intrusive load monitoring research," in *2nd Workshop on Data Mining Applications in Sustainability (SustKDD)*, p. 2012, 2011.

[18] J.-P. Zimmermann, M. Evans, J. Griggs, N. King, L. Harding, P. Roberts, and C. Evans, "Household electricity survey: A study of domestic electrical product usage," *Intertek Testing & Certification Ltd*, 2012.

[19] 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*, vol. 25, pp. 59–62, 2011.

[20] J. Kelly and W. Knottenbelt, "Uk-dale: A dataset recording uk domestic appliance-level electricity demand and whole-house demand," *ArXiv e-prints*, vol. 59, 2014.

[21] P. Y. Simard, D. Steinkraus, J. C. Platt, *et al.*, "Best practices for convolutional neural networks applied to visual document analysis.," in *ICDAR*, vol. 3, pp. 958–962, Citeseer, 2003.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[23] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Advances in Neural Information Processing Systems*, pp. 1324–1332, 2010.

[24] S. Seguí, O. Pujol, and J. Vitria, "Learning to count with deep object features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 90–96, 2015.