Deep Latent Generative Models For Energy Disaggregation

Gissella Bejarano
SUNY Binghamton
gbejara1@binghamton.edu

David DeFazio
SUNY Binghamton
ddefazi1@binghamton.edu

Arti Ramesh
SUNY Binghamton
artir@binghamton.edu

Abstract

Thoroughly understanding how energy consumption is disaggregated into individual appliances can help reduce household expenses, integrate renewable sources of energy, and lead to efficient use of energy. In this work, we propose a deep latent generative model based on variational recurrent neural networks (VRNNs) for energy disaggregation. Our model jointly disaggregates the aggregated energy signal into individual appliance signals, achieving superior performance when compared to the state-of-the-art models for energy disaggregation, yielding a 29% and 41% performance improvement on two energy datasets, respectively, without explicitly encoding temporal/contextual information or heuristics. Our model also achieves better prediction performance on low-power appliances, paving the way for a more nuanced disaggregation model. The structured output prediction in our model helps in accurately discerning which appliance(s) contribute to the aggregated power consumption, thus providing a more useful and meaningful disaggregation model.

Introduction

Designing machine learning models for smart energy consumption is an important research problem, having a tremendous impact on society. A crucial sub-problem in facilitating smart energy consumption is being able to accurately disaggregate energy signals into their component appliance signals. This process is also known as energy disaggregation/non-intrusive load monitoring (NILM). This exercise provides residents with an accurate view and understanding of their energy consumption and can potentially help in reducing the peak energy consumption and facilitating efficient usage and conservation of energy. Recent advances in variational inference for deep learning have resulted in more expressive deep generative models such as variational auto-encoders (VAEs) and variational recurrent neural networks (VRNNs) that possess the ability to encode continuous latent variables. These latent variables provide the models with a powerful layer of abstraction that captures the variations in the input data and helps in generating the output data. These models map the input sequence into continuous latent variables using an inference network (encoder), and then use the generative network (decoder) to reconstruct the input sequence by sampling from the latent variables. Chung et al. (2015) propose variational recurrent neural networks (VRNNs), which extend VAEs to model sequences by introducing high-level latent variables in RNNs. Deep generative models have achieved state-of-the-art performance in many sequence-to-sequence language tasks such as machine translation, paraphrase generation, and textual entailment, but have not been explored for the problem of energy disaggregation.

In this work, we present a novel deep generative architecture for disaggregation that leverages and adapts VRNNs to jointly disaggregate the total energy consumption into individual component appliance signals. Our proposed approach learns the abstraction of the aggregated energy consumption over latent variables at training time and then generates all the individual appliance signals jointly by sampling from the latent variables at test time. Hence, at test time our model only depends on the aggregated signal and the latent variable abstractions learned during training and does not depend on contextual information and appliance data from previous time steps, making it a meaningful model for energy disaggregation.

Specifically, we make the following contributions:

1. We present a novel deep generative architecture for performing sequence-to-many-sequence prediction (aggregated consumption to appliance consumptions) needed for energy disaggregation by leveraging and adapting variational recurrent neural networks (VRNNs). Our model generates continuous power consumption signals as opposed to state-of-the-art approaches that model consumption through discrete appliance states.

2. We model the structure among the different appliances in a household by jointly predicting each of them at the same time from the aggregated signal. We cast the different appliance energy signatures as a structured prediction problem, modeling the structure among the different appliance energy consumption signals over time, to effectively represent and reason about their dependence.

3. Our model achieves a performance improvement of *29%* and *41%* for the REDD and Dataport datasets, respectively, when compared to two recent state-of-the-art energy disaggregation approaches that use exten-

sive additional past temporal and contextual information (Tomkins, Pujara, and Getoor 2017; Shaloudegi et al. 2016). Further, our model achieves a superior prediction performance on low power consuming appliances, which are harder to predict and are often ignored by most existing approaches.

4. Through qualitative analysis, we demonstrate that our models can achieve a superior disaggregation for both high and low energy consumption states and accurately discerns which appliance(s) contribute to the aggregated power consumption, thus providing a more useful and meaningful disaggregation model.
5. We demonstrate the extensibility of our model in predicting individual appliance consumption on previously unseen data by testing on a building that is left out while training. We observe that our model achieves a superior prediction performance on two buildings in REDD, thus making it potentially extensible to new datasets.

Related Work

Hart et al. (Hart 1992) was the first to introduce the problem of energy disaggregation. Perhaps the most popular approach for energy disaggregation is using factorial hidden Markov models (FHMMs) (Ghahramani and Jordan 1997), which generalize HMMs by using a distributed representation and its variants (Kim et al. 2011; Kolter and Jaakkola 2012; Parson et al. 2012; Johnson and Willsky 2013). Shaloudegi et al. (2016) propose a scalable algorithm for this problem that extends FHMMs. Supervised machine learning models such as Support Vector Machines (SVMs) and k-Nearest Neighbors (k-NNs) and unsupervised models that use prior appliance models have also been applied to this problem (Altrabalsi et al. 2016; Barker 2014; Faustine et al. 2017; Makonin et al. 2016). Some other models have relied on other information apart from the aggregated consumption to model relationships with user's behavior and climate (Li and Zha 2016; Tomkins, Pujara, and Getoor 2017; Zhong, Goddard, and Sutton 2015). Tomkins et al. (2017) propose a structured probabilistic framework for energy disaggregation. Recent advances in deep learning have spurred deep-learning based energy disaggregation models (Kelly and Knottenbelt 2015; Lange and Bergés 2016; Barsim, Mauch, and Yang 2018; do Nascimento 2016; Zhang et al. 2018; Huss 2015; Zhang et al. 2018; Huss 2015; Batra et al. 2018).

In this work, we propose a deep latent generative model based on VRNNs that combines the advantages of the modeling complexity of deep neural networks and the rich representational power of latent variables in probabilistic models such as FHMMs. Our model learns to predict all individual appliance signals *jointly* from the aggregated signal. We compare our approach to two recent state-of-the-art approaches for energy disaggregation: a) ADMM-RR, a scalable variant of FHMMs (Shaloudegi et al. 2016), and b) Tomkins et al.'s (2017) joint probabilistic approach to energy disaggregation, and show that our approach achieves superior prediction performance.

Deep Latent Generative Models for Energy Disaggregation

In this section, we describe the energy disaggregation problem and the suitability of VRNNs for the same. Then, we present our deep latent generative energy disaggregation architecture.

Energy Disaggregation Problem

The problem of disaggregation is to calculate the energy consumption of individual component appliances given the total aggregated power consumption. Let $\mathbf{x} = (x_1, x_2, ..., x_T)$ be the aggregated energy consumption of a house over T time steps, where $x_t \in \mathbb{R}_+$. Let I be the number of appliances. The individual energy consumption of appliance i is denoted by $\mathbf{y^i} = (y_1^i, y_2^i, ..., y_T^i)$, where $y_t^i \in \mathbb{R}_+$. Consequently the aggregated energy signal at a given time can also be expressed as $x_t = \sum_{i=1}^{I} y_t^{(i)}$. We use y_t to denote the consumption time t for all the appliances: $y_t = \{y_t^1, y_t^2, ..., y_t^I\}$. *Our goal in this work is to develop a deep latent generative energy disaggregation framework that can learn to infer the continuous-valued appliances' consumption given the aggregated energy consumption.*

Variational Recurrent Neural Networks

Variational Recurrent Neural Networks (VRNNs) (Chung et al. 2015) are a recently developed deep neural network architecture that introduce latent variables and temporal dependencies between them in the different time steps in the RNN architecture. The core of a VRNN is a variational auto-encoder (VAE) (Kingma and Welling 2014; Rezende, Mohamed, and Wierstra 2014). VAEs and VRNNs are variants of autoencoders (AE) and recurrent neural networks (RNNs) that encode latent variables and probabilistic transition functions. The principal difference between VAE and VRNN is that VRNN models the dependencies between latent variables across subsequent time steps, thus providing us with the ability to accurately abstract highly non-linear dynamics in sequential data. Since the prior distribution at timestep t is dependent on all the preceding inputs via the RNN hidden state h_{t-1}, the introduction of temporal structure in the prior distribution is expected to improve the representational power of the model. We first discuss the suitability of VRNNs for the energy disaggregation problem and then present our deep generative architecture.

Why are VRNNs suitable for the energy disaggregation problem? As Chung et al. (2015) note, VRNNs are best suited for modeling highly variable and highly structured (having a high signal-to-noise ratio) sequential data. Highly variable data exhibits high sudden variations that vanilla RNNs do not accurately represent. The deterministic nature of transition functions in RNNs limit their capability in modeling variability in the outputs. The presence of latent variables in VRNNs allows them to represent latent state spaces similar to models such as hidden Markov models (HMMs) and Kalman filters in a deep neural network architecture such as RNNs, thus achieving the combined benefits of both these classes of models.

Energy consumption signals are highly structured, i.e., they have a high signal to noise ratio; the variations in the data are due to signal itself rather than noise. Thus, the presence of structured output functions in VRNNs along with their ability to represent complex non-linear data make them ideal for modeling this domain.

The structured output functions present in VRNNs aid the joint prediction of disaggregated appliance signals from the aggregated consumption. Kelly et al. (2015) use RNNs for the energy disaggregation problem, but their model does not disaggregate all appliance signals at once. Instead, they train a separate model for each appliance. Due to the lack of probabilistic transitions between latent variables and structured output functions, this approach fails to capture the dependencies between the different appliance signals and thus lacks the ability to accurately identify the contributing appliance signals in an aggregated signal.

In the energy disaggregation problem, usually power consumption is mapped to discrete appliance states (Shaloudegi et al. 2016; Tomkins, Pujara, and Getoor 2017). This, however, ignores the fine-grained variations in the signals. The deep structured construction of our model and the presence of latent variable abstractions and probabilistic transitions between them provide us with the ability to model the exact consumption of appliances as continuous values and detect fine-grained variations in the signals. Since we do not approximate signals into consumption states and model the exact continuous values, our approach requires minimal preprocessing and is able to model these minute variations.

VRNN-DIS-ALL: A Deep Generative Energy Disaggregation Framework

We bring out the modeling power of VRNNs by adapting them to disaggregate individual appliance signals jointly from the aggregated power consumption signal. In the following sections, we present the generative process, inference, and learning in our model, VRNN-DIS-ALL. We also highlight the adaptations to the original VRNN for the energy disaggregation problem.

Generation The VRNN contains a VAE in each time step but the prior on the latent variable follows a distribution that is conditioned on the hidden state at time $t-1$, h_{t-1}. We augment the prior distribution to include both h_{t-1} and the aggregated consumption at time t, denoted by x_t. Hence, the random variable z_t follows the distribution:

$$z_t \sim N(\mu_{0,t}, diag(\sigma^2_{0,t})) \tag{1}$$

where, $[\mu_{0,t}, \sigma_{0,t}]$ denote the parameters of the distribution $\phi^{prior}_{\tau}(h_{t-1}, x_t)$. While for the generation task described in Chung et al. (2015), the prior distribution only depends on h_{t-1}, we adapt it to include the aggregated signal as we are interested in generating the disaggregated appliance signal from the aggregated signal. Next, y_t (disaggregated signal) is generated given z_t and h_{t-1} from the distribution:

$$y_t|z_t \sim N(\mu_{y,t}, diag(\sigma^2_{y,t})) \tag{2}$$

where, $[\mu_{y,t}, \sigma_{y,t}] = \phi^{dec}_{\tau}(\phi^{z}_{\tau}(z_t), h_{t-1})$. Chung et al. (2015) note that ϕ^{prior}_{τ} and ϕ^{dec}_{τ} can be any highly flexible functions and are essential for learning complex dependencies. In our models, ϕ^{prior}_{τ} and ϕ^{dec}_{τ} are neural networks with one-hidden layer with standard activation functions. The hidden layer has a hyperbolic tangent (*tanh*) and the output layers for $\mu_{y,t}$ and $\sigma_{y,t}$ have linear and softplus activations, respectively. The RNN hidden state calculation is given by

$$h_t = f(\phi^{x}_{\tau}(x_t), \phi^{y}_{\tau}(y_t), \phi^{z}_{\tau}(z_t), h_{t-1}) \tag{3}$$

where, f is the transition function between hidden states. The feature extractors, ϕ^{x}_{τ}, $\phi^{y}_{\tau}(y_t)$, and ϕ^{z}_{τ}, can be any expressive function. We use a 1-hidden layer neural network for the same. The learning problem is to learn the prior distribution, $\phi^{prior}_{\tau}(h_{t-1}, x_t)$, to be as close as possible to the approximate posterior $\phi^{enc}_{\tau}(\phi^{x}_{\tau}(x_t), \phi^{y}_{\tau}(y_t), h_{t-1})$.

Inference At training time, VRNN works as an encoder, learning the approximate posterior as a function of x_t, y_t and h_{t-1}.

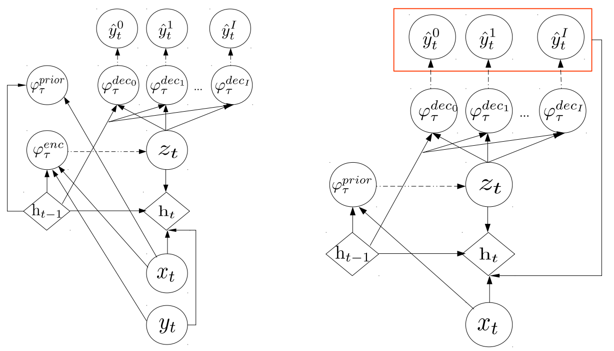
$$z_t|x_t, y_t \sim N(\mu_{z,t}, diag(\sigma^2_{z,t})) \tag{4}$$

where, $[\mu_{z,t}, \sigma_{z,t}]$ denote the parameters of the distribution $\phi^{enc}_{\tau}(\phi^{x}_{\tau}(x_t), \phi^{y}_{\tau}(y_t), h_{t-1})$. In addition to the feature extractors from x_t and z_t, we also include $\phi^{y}_{\tau}(y_t)$ that extracts the features of the disaggregated signal y_t at training time. Inference at training time is done by sampling z_t from this approximated posterior distribution. At test time, z_t is sampled from the learned prior distribution that is learned during training. This difference in the z_t can be appreciated in Figure 1 where we can see how the distribution ϕ^{prior}_{τ} replaces the encoder distribution ϕ^{enc}_{τ}.

Learning Learning is performed by minimizing the sum of two components: distance between the posterior and the prior distribution and the log-likelihood of the output. In the first term in Equation 5, we minimize the Kullback-Leibler divergence distance (KL divergence) between the approximate posterior in Equation 4 (denoted by q in Equation 5) and the prior distribution (denoted by p in Equation 5), where z_t depends only on aggregated signal ($x \leq t$) and the latent variable states at previous time steps ($z < t$). The second term captures the negative log-likelihood of the output distribution from which we sample y_t.

$$\begin{aligned} KL(q(z_t|x \leq t, y \leq t, z < t)||p(z_t|x < t, z < t)) \\ +logp(y_t|z \leq t, x < t) \end{aligned} \tag{5}$$

Training At training time, our goal is to learn an approximate function that is very similar to the conditional distribution $p(z|y)$ by minimizing the KL divergence between the prior distribution (ϕ^{prior}_{τ}) and the approximate posterior or the encoder distribution (ϕ^{enc}_{τ}) (Figure 1a). We follow a curriculum learning strategy as proposed by Bengio et al. (2015) that involves gradually migrating during training from considering the ground truth to the output predicted by the model in the previous step in order to bridge the gap in inference between training and testing. The scheduled sampling algorithm used by this learning strategy will decide at training time whether to sample from the ground truth (y_t) or from the predictions generated by the model ($\hat{y}_t$). In our models, we use an inverse sigmoid decay. It is defined as: $p_i = k/(k + exp(i/k))$ where, p_i is the sampling probability and $k >= 1$ gives the speed of convergence. This probability is calculated at each time step.



(a) VRNN-DIS-ALL at training (b) VRNN-DIS-ALL at test

Figure 1: Graphical illustrations of VRNN-DIS-ALL training to reconstruct disaggregated appliance signals from the aggregated and disaggregated signals and as a generative model of disaggregated appliance signals from only the aggregated signal at test time.

Testing At test time, we only input the aggregated energy consumption information x_t. We no longer use the encoder distribution but the parameters of the learned prior distribution to sample the latent variables z_t, i.e., $z_t \sim N(\mu_{0,t}, diag(\sigma^2_{0,t}))$. Then, we calculate the parameters of the distribution of each appliance using $y^i_t|z_t \sim N(\mu^i_{y,t}, diag({\sigma^i_{y,t}}^2))$, where $[\mu^i_{y,t}, \sigma^i_{y,t}]$ is now calculated from the learned prior distribution. We calculate the next recurrent hidden layer h_t as a function of feature extractor neural networks for x_t, $\hat{y}_t$, and z_t, and previous hidden state h_{t-1}, i.e., $f(\phi^x_\tau(x_t), \phi^z_\tau(z_t), \phi^y_\tau(\hat{y}_t), h_{t-1})$. Note that here we use the predicted $\hat{y}$ instead of the actual y (Figure 1b).

Implementation Details We develop our model[1] on the original VRNN implementation (Chung et al. 2015) in Theano. Figure 2 captures the model architecture. It shows the different hidden layers, number of nodes in the hidden layers, identifies the components corresponding to the hidden layer, and captures the interactions between them for one iteration of training from time $t-1$ to t. The name of each component and the activation function applied to nodes in that hidden layer is mentioned at the top and the number of nodes is indicated in the bottom of each hidden layer. The architecture shows the distribution from where z_t will be sampled at training time (from the encoder, marked in green) and at inference/test time (from the prior, marked in red). The weight matrices of all layers are randomly initialized using a uniform distribution. The LSTM-cell diagonal matrix that captures the interaction between the recurrent states h_{t-1} and h_t is initialized randomly from a normal distribution ensuring its orthogonality. The initial hidden state of the recurrent neural network is initialized to 0.

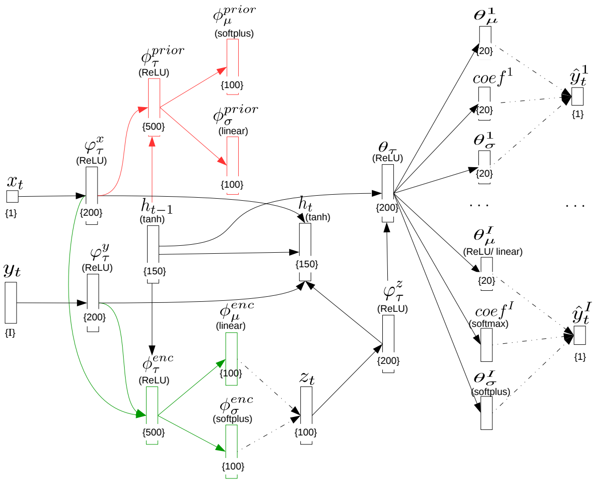


Figure 2: Architecture of the VRNN-DIS-ALL model. Solid lines represent fully connected layers and dashed lines represent the sampling process.

We experiment with different activation functions for θ_μ and find that ReLU activation function for θ_μ works better for some buildings while for others the linear activation function works better. For θ_σ and $coef^1$, we apply a *softmax* and *softplus* activation functions, respectively. These parameters are calculated for each appliance, so the final layer has as many Gaussian mixture models (GMMs) as appliances.

Experimental Evaluation

We conduct experiments to answer the following questions:

1. How well do our deep generative models perform in energy disaggregation?
2. Are our models able to effectively identify which appliance(s) are contributing to the aggregated consumption?

Model	Disaggregation Representation	Temporal Dependencies	Context/ Heuristics
ADMM-RR (Shaloudegi et al. 2016)	Discrete states	Encoded	✓
INTERVAL (Tomkins et al. 2017)	Discrete states	Encoded	✓
INSTANCE (Tomkins et al. 2017)	Discrete states	Encoded	✓
CONTEXT (Tomkins et al. 2017)	Discrete states	Encoded	✓
VRNN-DIS-ALL (our approach)	Continuous	Learned	✗

Table 1: A comparison table between our model and the state-of-the-art energy disaggregation approaches

In this section, we present results from our experimental evaluation to answer the above-mentioned questions on two well-known real-world energy disaggregation datasets. We demonstrate the efficacy of our models by comparing them with two recent state-of-the-art energy disaggregation approaches: a) ADMM-RR (Shaloudegi et al. 2016), and b) Interval, Instance, and +Context models from Tomkins et al. (2017). Table 1 gives a comparison of our approach with the state-of-the-art energy disaggregation models. Our

[1] https://bitbucket.org/gissemari/disaggregation-vrnn

approach uses a continuous value representation, does not explicitly encode any domain-specific variables or capture any dependencies among them, and does not require any additional contextual information (such as temperature of the day, day of the month/year, user-specific contextual information). Our model automatically learns these dependencies from training data. In our experiments, we demonstrate that our models outperform ADMM-RR across most appliances in both the datasets and outperforms Tomkins et al.'s best model on one dataset and achieves comparable performance on another despite using no temporal, domain-related, or contextual information.

We present two metrics of evaluation for both the datasets: i) mean absolute error (MAE), and ii) percentage of total energy estimated by each appliance compared to percentage of total energy in the original data. The MAE is calculated by computing the absolute value of the difference between the predicted disaggregated appliance consumption $(\hat{y_t})$ and the actual consumption (y_t). Percentage of energy estimated is calculated by taking the ratio of predictions associated with the appliance to original aggregated signal. This percentage is compared with the actual percentage of energy consumption of the appliance in the aggregated energy consumption. We evaluate our percentage predictions in the following ways: i) first, we compare the actual percentage numbers between our predictions and the actual data, ii) second, we compute the percentage/range of error between the predicted and the actual by taking the ratio of the difference in the percentages with the actual percentage, and iii) third, we compare our deviation in percentages (percentage of error) to the deviation in percentages reported for the same building by Tomkins et al.

Datasets

We evaluate our model on two real-world energy datasets: i) Pecan Street Inc. Dataset (DataPort) (dat 2016), and ii) Reference Energy Disaggregation Dataset (REDD) (Kolter and Johnson 2011). These datasets have been used in several previous works (Tomkins, Pujara, and Getoor 2017; Shaloudegi et al. 2016; Makonin et al. 2016).

DataPort The Pecan Street dataset (DataPort) consists of energy consumption readings at 1-minute and 1-hour intervals. We evaluate on the finer-grained 1-minute readings. As there are missing values, we work on the same subset of buildings (2859, 3413, 6990, 7951, 8292) that Tomkins et al. (2017) use in their work. We consider data for the following appliances: *air conditioner*, *furnace*, *refrigerator*, *dishwasher*, *kitchen outlet*, *dryer*, *microwave*, and *clothes washer*.

REDD The REDD dataset contains data for 10 houses from the greater Boston area for approximately two months. We again consider the same five houses (houses 1, 2, 3, 4, and 6) that Tomkins et al. (2017) and Makonin et al. (2016) consider so that we can make a fair comparison. We also consider the same four appliances for houses 1, 2, and 3: *refrigerator*, *dishwasher*, *light*, *microwave*. For house 6 we exclude *microwave* as the data for that appliance is unavailable. We use the non-intrusive load monitoring toolkit (Batra et al. 2014) to get a sampling rate of every 6 or 60 seconds.

Appliance	**Building**					
	2859	**6990**	**7951**	**8292**	**3413**	**AVG/Appliance**
Air	9.50	199.50	152.50	98.50	64.00	104.80
Furnace	40.50	110.50	59.00	56.50	32.50	59.80
Refrigerator	32.50	70.00	77.50	60.00	71.50	62.30
Clothes washer	1.50	6.00	24.00	8.50	2.50	8.50
Dryer	4.00	52.00	33.00	78.50	35.50	40.50
Dish washer	1.00	8.00	14.50	25.00	9.50	11.60
Kitchenapp	1.00	3.00	14.00	17.50	1.00	7.30
Microwave	10 .00	12.50	40.00	9.00	6.00	15.50
AVG/Building	12.50	57.69	51.81	44.13	27.81	**38.79**

Table 2: VRNN-DIS-ALL results on DataPort showing the MAE for each appliance for the five buildings.

Data Preprocessing

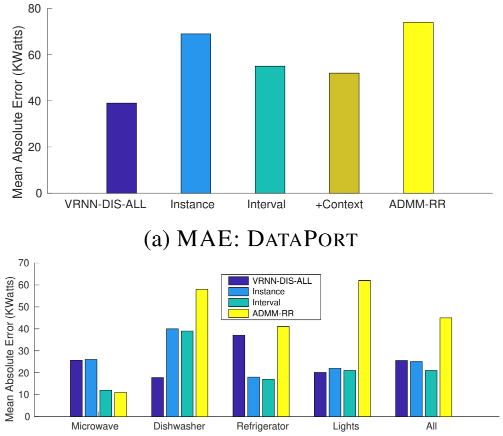
To preprocess the data for our model, we first determine the minimum activation threshold for each appliance in each dataset. Then, we use a non-overlapping sliding window on the entire original time series data to construct sequences of fixed length from them. From these sequences, we filter the ones where at least one data point in the sequence is greater than the minimum threshold activation for each appliance. We treat each sequence as one data instance. We split the total number of instances into training, testing, and validation sets in the ratio 50%:25%:25%, respectively. We record the performance metrics in the validation set every ten epochs to detect and prevent overfitting.

We construct batches of instances (which we refer to as *mini-batch*) and train the model for many epochs for each mini-batch. This enables the model to see a smaller number of instances for a longer training period, enabling it to model the structural dependencies in the data. We use 5-30 mini-batches. We report the average scores from three different train-test-validation splits across both datasets. Note that our approach uses very minimal pre-processing and domain knowledge when compared to the existing state-of-the-art approaches.

Energy Disaggregation Results on DataPort

Table 2 shows the MAE of each appliance in each building in DataPort dataset. Our model is able to achieve low MAE for appliances that consume higher energy in average such as *clothes washer* and *air conditioner*. The first one shows a MAE of 1.5, 6, 8.5, and 2.5 in buildings 2859, 6990, 8292 and 3413, respectively and the *air conditioner* obtains a MAE of 9.5 for building 2859. In addition to that, appliances which consume less energy on average such as *dishwasher*, *kitchen appliance*, and *microwave* show an average MAE of less or equal than 15.5 among all buildings. It is interesting to note that Tomkins et al.'s prediction performance of appliance states for appliances that consume lesser power on average and are intermittent is lower as indicated by their lower values of precision, recall and F1 scores. Thus, our model is able to discern patterns of consumption in both kinds of appliances and hence perform a more accurate disaggregation.

Figure 3a shows the comparison of average MAE value across the buildings and appliances between our model VRNN-DIS-ALL and two existing state-of-the-art approaches. Since we consider the same set of buildings and appliances, we make a direct comparison to the results presented by Tomkins et al. (2017). We observe that our model achieves *29%* performance improvement in MAE over the +CONTEXT model (Tomkins et al.'s best model) and *41%* improvement over ADMM-RR. It is important to note that our model achieves this performance improvement without any contextual information. The deep nature of the model and the presence of neural network feature extractors help in extracting complex features and learning structural dependencies among them. This eliminates the necessity to encode domain-specific information and their relationships as in existing probabilistic energy disaggregation approaches. Hence, our approach requires less manual effort and can scale easily to new datasets without the need for careful encoding of graphical structure among variables.



(a) MAE: DATAPORT

(b) MAE: REDD

Figure 3: MAE comparing our proposed model VRNN-DIS-ALL with existing state-of-the-art models (Interval, Instance, +Context, and ADMM-RR)

Figure 4 gives the percentage of total energy consumed by the appliance as predicted by our model compared with the actual percentage of total energy consumed by the appliance in the original data. We group the appliances into *air conditioner* (air), *furnace*, *refrigerator*, *dryer*, and *others*, to enable an easy comparison to Tomkins et al.'s percentage calculations. Comparing the predicted percentage of total energy with the actual for *air conditioner* across all buildings, we observe that our model predicts within 4% of the actual percentage of energy consumed by the appliance for 4 out of 5 buildings. Similarly, for *dryer*, our model's predictions lie within 11% for 4 out of 5 buildings, and for *furnace*, our model's predictions lie within 6% for 3 out of 5 buildings. Comparing the percentages for building 3413 with the percentages reported by Tomkins et al., we observe that our model's percentage prediction for *air conditioner* deviates by only 3.6% from the actual percentage, while theirs deviates by 7.3%. Similarly, comparing the percentages for *furnace* we observe that ours deviates by 1.5% while theirs deviates by 10%. For the rest of the appliances, our model achieves comparable differences in percentages between the predicted and the actual values to their model.

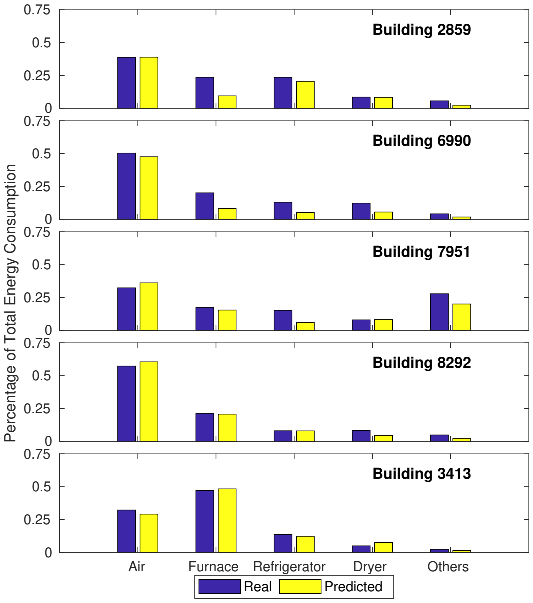


Figure 4: Percentage of total energy consumption of each appliance for Dataport homes

Energy Disaggregation Results on REDD

Figure 3b gives the comparison for average MAE of each appliance across buildings between VRNN-DIS-ALL and existing state-of-the-art approaches. Here, we only compare against INTERVAL and INSTANCE models from Tomkins et al. as the +CONTEXT model cannot be used due to absence of contextual information in the dataset. We observe that our model achieves superior performance on *dishwasher* and *lights*, which are harder to predict due to their unpredictability. We get performance improvements of 69% and 68%, respectively, over ADMM-RR and 56% for dishwasher over Tomkins et al. For the other appliances: *microwave* and *refrigerator*, we achieve comparable performance to one of the existing approaches. Comparing our overall MAE averaged over all buildings and all appliances with ADMM-RR, we observe that VRNN-DIS-ALL achieves a performance improvement of 41%. Our overall MAE is comparable to Tomkins et al.'s models, despite having no careful encoding of domain-specific temporal, contextual, and structural dependencies using graphical templates, paving the way for a model that can be extended easily to other settings.

Figure 5 compares the percentage of energy consumption by each appliance with the actual energy consumption percentages. Our actual percentage values for *refrigerator* differ by less than 1% for buildings 1 and 6. We observe a similar trend for *light*, where VRNN-DIS-ALL's predictions achieve the exact same percentage for building 6 and only an actual difference in percentage values of $< 4\%$ for building 1. Again, comparing the percentages for building 3 with

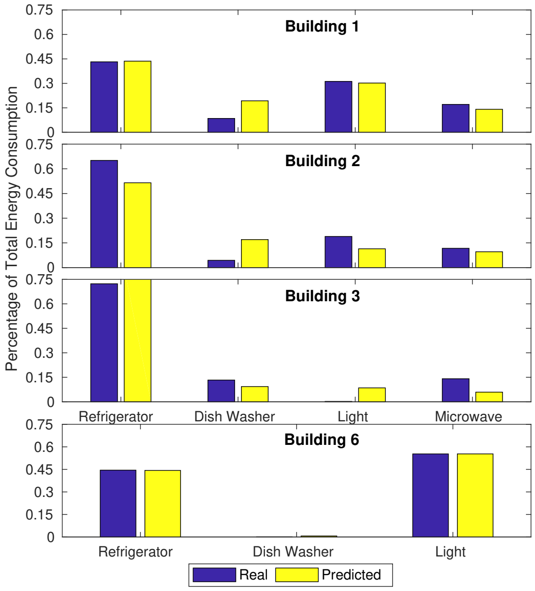


Figure 5: Percentage of total energy consumption of each appliance for REDD homes

the percentages reported by Tomkins et al., we observe that our model's percentage prediction for *refrigerator* deviates by 13% from the actual percentage, while theirs deviates by 16.5%. Similarly, for *dishwasher*, our predictions deviate by 30% while Tomkins et al.'s deviate by 49%.

Further, the latent variable abstractions help our model discern which appliance(s) contribute to the aggregated power consumption and distinguish between appliance signatures, demonstrating the ability to perform blind source separation (Pal et al. 2013). In Figure 6, we show an example of disaggregation for REDD. We observe that the aggregated energy consumption (first subfigure from top) is significantly contributed by *light* and *refrigerator*. Our model accurately detects both these phenomena in the predictions by identifying the presence of two peaks in this time period and the respective appliances responsible for them. These qualitative results demonstrate that our model is indeed learning to split the aggregated energy consumption into its component appliance signals.

Training	Building				
	1	2	3	6	AVG
Same building	32.00	31.63	25.00	7.50	**25.17**
Unseen building	40.00	25.25	17.75	71.33	**35.54**

Table 3: VRNN-DIS-ALL MAE results on different data seen for training the REDD dataset

Testing on Unseen Data We evaluate the performance of our model training on all buildings leaving one building out and testing on that building. From Table 3, we can see that the MAEs for buildings 2 and 3 improve while building 1 gets a comparable MAE. The overall MAE across all buildings and appliances is also comparable to the result obtained when training on the same building with a difference of only ~ 10 in MAE and still achieving a superior prediction performance than ADMM-RR, illustrating the ability of our models to be extensible across buildings of the same dataset.

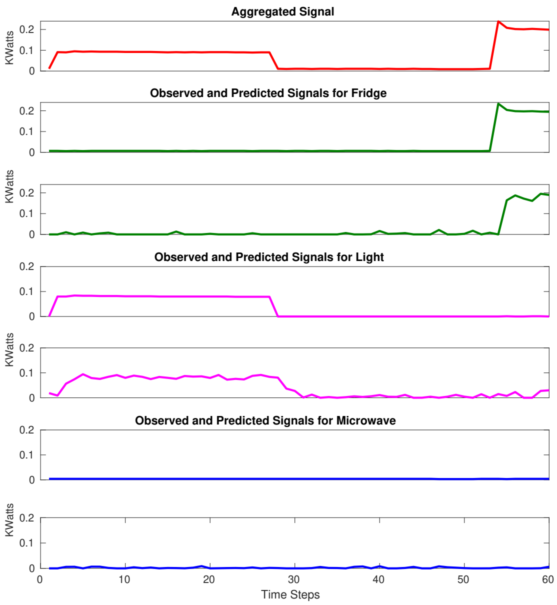


Figure 6: Figures showing an example disaggregation by VRNN-DIS-ALL in REDD using aggregated and disaggregated ground truth and predicted signals.

Discussion

In this paper, we presented a novel deep generative framework that adapts a very recently developed generative model, VRNNs for energy disaggregation. We demonstrated that our model is capable of performing sequence-to-many-sequence prediction to disaggregate the aggregated energy consumption into individual appliance consumption signals. We further demonstrated that our models are capable of achieving superior performance in two well-known real-world energy disaggregation datasets DataPort and REDD, achieving 29% and 41% improvement in MAE from the existing state-of-the-art approaches. We also demonstrated the capability of our framework in accurately predicting energy consumption of appliances that consume less power and have no discernible repeating pattern, thus paving the way toward a fine-grained and informed energy disaggregation. Further, the latent variable abstractions help in achieving good prediction performance on previously unseen data. There are many exciting future directions. The generative nature of our models facilitates generating synthetic data that captures the minute variations in the signal. The latent variable abstractions can potentially be tuned to achieve good prediction performance across energy signals from different locations.

References

Altrabalsi, H.; Stankovic, V.; Liao, J.; and Stankovic, L. 2016. Low-complexity energy disaggregation using appliance load modelling. *AIMS Energy* 4(1):884–905.

Barker, S.; Kalra, S.; Irwin, D.; and Shenoy, P. 2014. Empirical characterization, modeling, and analysis of smart meter data. *IEEE Journal on Selected Areas in Communications* 32(7):1312–1327.

Barker, S. K. 2014. *Model-driven analytics of energy meter data in smart homes*. Ph.D. Dissertation, University of Massachusetts - Amherst.

Barsim, K. S.; Mauch, L.; and Yang, B. 2018. Neural network ensembles to real-time identification of plug-level appliance measurements. In *Proceedings of the 2016 NILM workshop*.

Batra, N.; Kelly, J.; Parson, O.; Dutta, H.; Knottenbelt, W.; Rogers, A.; Singh, A.; and Srivastava, M. 2014. Nilmtk: an open source toolkit for non-intrusive load monitoring. In *Proceedings of the 5th international conference on Future energy systems*, 265–276. ACM.

Batra, N.; Jia, Y.; Wang, H.; Whitehouse, K.; et al. 2018. Transferring decomposed tensors for scalable energy breakdown across regions. In *AAAI*.

Bengio, S.; Vinyals, O.; Jaitly, N.; and Shazeer, N. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*.

Chung, J.; Kastner, K.; Dinh, L.; Goel, K.; Courville, A.; and Bengio, Y. 2015. A recurrent latent variable model for sequential data. In *Proceedings of the International Conference on Neural Information Processing Systems*, NIPS.

2016. Pecan street inc., dataport.

do Nascimento, P. P. M. 2016. *Applications of Deep Learning Techniques on NILM*. Ph.D. Dissertation, Universidade Federal do Rio de Janeiro.

Faustine, A.; Mvungi, N. H.; Kaijage, S.; and Michael, K. 2017. A survey on non-intrusive load monitoring methodies and techniques for energy disaggregation problem.

Ghahramani, Z., and Jordan, M. I. 1997. Factorial hidden markov models. *Machine Learning* 29(2):245–273.

Hart, G. W. 1992. Nonintrusive appliance load monitoring. *Proceedings of the IEEE* 80(12):1870–1891.

Huss, A. 2015. Hybrid model approach to appliance load disaggregation: Expressive appliance modelling by combining convolutional neural networks and hidden semi markov models.

Johnson, M. J., and Willsky, A. S. 2013. Bayesian nonparametric hidden semi-markov models. *J. Mach. Learn. Res.* 14(1):673–701.

Kelly, J., and Knottenbelt, W. 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, 55–64. New York, NY, USA: ACM.

Kim, H.; Marwah, M.; Arlitt, M.; Lyon, G.; and Han, J. 2011. *Unsupervised Disaggregation of Low Frequency Power Measurements*.

Kingma, D. P., and Welling, M. 2014. Auto-encoding variational bayes. *The International Conference on Learning Representations*.

Kolter, J. Z., and Jaakkola, T. 2012. Approximate inference in additive factorial hmms with application to energy disaggregation. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22, 1472–1482. La Palma, Canary Islands: PMLR.

Kolter, J. Z., and Johnson, M. J. 2011. REDD: A public data set for energy disaggregation research. In *Proceedings of the KDD Workshop on Sustainability (SustKDD)*.

Lange, H., and Bergés, M. 2016. The neural energy decoder: energy disaggregation by combining binary subcomponents. In *Proceeding of the 3rd International Workshop on Non-Intrusive Load Monitoring*.

Li, L., and Zha, H. 2016. Household structure analysis via hawkes processes for enhancing energy disaggregation. In *IJCAI*, 2553–2559.

Makonin, S.; Popowich, F.; Bajić, I. V.; Gill, B.; and Bartram, L. 2016. Exploiting hmm sparsity to perform online real-time nonintrusive load monitoring. *IEEE Transactions on Smart Grid* 7(6):2575–2585.

Pal, M.; Roy, R.; Basu, J.; and Bepari, M. S. 2013. Blind source separation: A review and analysis. In *International Conference Oriental COCOSDA held jointly with Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE)*, 1–5.

Parson, O.; Ghosh, S.; Weal, M.; and Rogers, A. 2012. Non-intrusive load monitoring using prior models of general appliance types. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 356–362.

Rezende, D. J.; Mohamed, S.; and Wierstra, D. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Shaloudegi, K.; György, A.; Szepesvári, C.; and Xu, W. 2016. Sdp relaxation with randomized rounding for energy disaggregation. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*.

Tomkins, S.; Pujara, J.; and Getoor, L. 2017. Disambiguating energy disaggregation: A collective probabilistic approach. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2857–2863.

Zhang, C.; Zhong, M.; Wang, Z.; Goddard, N. H.; and Sutton, C. A. 2018. Sequence-to-point learning with neural networks for non-intrusive load monitoring. In *The Thirty-Second AAAI Conference on Artificial Intelligence*, volume abs/1612.09106.

Zhong, M.; Goddard, N.; and Sutton, C. 2015. Latent bayesian melding for integrating individual and population models. In *Advances in neural information processing systems*, 3618–3626.