"I do not know": Quantifying Uncertainty in Neural Network Based Approaches for Non-Intrusive Load Monitoring

Vibhuti Bansal*†
bansal.vibhuti25@gmail.com
Bharati Vidyapeeth's College of Engineering, Delhi, India

Rohit Khoiwal*†
khoiwalrohit.16@gmail.com
Rajasthan Technical University India

Hetvi Shastri*‡
hshastri@umass.edu
University of Massachusetts Amherst USA

Haikoo Khandor
haikoo.ashok@iitgn.ac.in
IIT Gandhinagar India

Nipun Batra
nipun.batra@iitgn.ac.in
IIT Gandhinagar India

ABSTRACT

Non-intrusive load monitoring (NILM) refers to the task of disaggregating total household power consumption into the constituent appliances. In recent years, various neural network (NN) based approaches have emerged as state-of-the-art for NILM. In conventional settings, NN(s) provide point estimates for appliance power. In this paper, we explore the question - can we learn models that tell when they are unsure? Or, in other words, can we learn models that provide uncertainty estimates? We explore recent advances in uncertainty for NN(s), evaluate 14 model variants on the publicly available REDD dataset, and find that our models can accurately estimate uncertainty without compromising on traditional metrics. We also find that different appliances in their different states have varying performance of uncertainty. We also propose "recalibration" methods and find they can improve the uncertainty estimation.

CCS CONCEPTS

• **Computing methodologies → Machine learning**.

KEYWORDS

Neural networks, Non-Intrusive Load Monitoring, Bayesian analysis, Uncertainty, Calibration

ACM Reference Format:
Vibhuti Bansal, Rohit Khoiwal, Hetvi Shastri, Haikoo Khandor, and Nipun Batra. 2022. "I do not know": Quantifying Uncertainty in Neural Network Based Approaches for Non-Intrusive Load Monitoring. In *The 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '22), November 9–10, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3563357.3564063

*These authors contributed equally to this research.
†Work done as an intern at IITGn
‡Work done as an IITGn student

BuildSys '22, November 9–10, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9890-9/22/11…$15.00
https://doi.org/10.1145/3563357.3564063

1 INTRODUCTION

Non-intrusive load monitoring [9], or NILM, is the technique of decomposing overall power consumption into constituent appliances. Prior studies [5] suggest that providing appliance-wise energy consumption can help users potentially reduce their energy consumption by up to 15%.

Since the seminal work on NILM by George Hart [9], a variety of algorithms have been proposed in the recent past, including, but not limited to, time-series models such as additive factorial hidden Markov models [15], discriminative sparse coding [14], graph signal processing [10]. In 2015, Kelly et al. [13] proposed the application of neural networks for NILM. Since then, several neural network-based approaches for NILM have been proposed [12, 20, 24].

Conventional NN approaches for NILM provide point estimates, i.e. they may say that the fridge power consumption at 10 AM is 150 Watts. These conventional NN approaches do not quantify uncertainty. In contrast, a method that can quantify uncertainty may say that the fridge power at 10 AM is normally distributed with a mean of 150 Watts and a standard deviation of 5 Watts. However,if the model predicted that the fridge power is normally distributed with a mean of 150 Watts but a high standard deviation of 50 Watts, it means our model is unsure. The application designer or decision maker can factor in the uncertainty in predictions before deciding.

Recent literature has looked into methods of quantifying uncertainty in prediction from neural networks [3, 7, 22]. Such methods have been employed in various applications, including but not limited to medical imaging[18, 23]. A conventional approach would apply Bayesian analysis by putting a prior distribution over all the weights of the NN and then computing the posterior over the weights and the predictive distribution. However, such an approach would be computationally intractable [3]. Thus, more recently, various approximate inference methods have been proposed for quantifying the uncertainty in NNs. These methods include heteroskedastic NNs where we modify the architecture for regression to include two output nodes (one for the mean and one for the variance) instead of one output node. Other methods create an ensemble of NNs and combine the predictions from the individual models to obtain predictive uncertainty.

Prior literature has proposed a metric called expected calibration error and reliability diagrams (or calibration curves) to quantify the "goodness" of the predicted uncertainty (often also called how well a model is calibrated). We explain a well-calibrated model with

an example. Suppose our model's output is a normal distribution's mean (μ) and standard deviation (σ). Now, a 95% credible interval (CI) would correspond to 2σ, and in a well-calibrated model, 95% of the data points (ground truth) would lie within the predicted $\mu \pm 2\sigma$.

In this paper, we implement a total of 14 such model variants over the state-of-the-art NNs for NILM and evaluate these on three appliances on the publicly available REDD dataset [16]. We also propose a "re-calibration" method to improve the uncertainty quantification from our models. We now summarise the main questions and their answers that we explore in this paper:

(1) Do NNs with uncertainty achieve comparable error on conventional metrics to the baselines?
   (a) We find that we can achieve comparable or better performance on conventional metrics while additionally incorporating the notion of uncertainty.
(2) Are certain appliances or appliance states more prone to poor calibration?
   (a) We find that sparsely used appliances (like a dishwasher) have poor calibration compared to regularly used appliances (like a fridge).
(3) Can recalibration improve model uncertainty?
   (a) We find that for most of our models, our proposed recalibration scheme can improve the quantification of model uncertainty.

The rest of the paper is structured as follows. First, we discuss the methods of incorporating uncertainty in neural networks and various Bayesian approximation methods in Section 2. We also discuss quantification of predictive uncertainty and recalibration method. In Section 3, we outline the Seq2Point [3.1] and Bi-LSTM with attention[3.2] architectures which are state-of-the-art architectures for NILM. We discuss the evaluation in Section 4. We analyse the results in Section 4.4. After that, in Section 5, we go over prospective directions for this work before concluding in Section 6.

2 UNCERTAINTY IN NEURAL NETWORKS

We now discuss different architectures and approximate inference technique to estimate uncertainty in neural networks. We summarise these techniques and architectures in Figure 1. We direct the reader to recent surveys for an in-depth discussion on the different methods for estimating uncertainty in neural networks. [7, 22]

2.1 Homoskedastic Neural Networks

Homoskedasticity in the context of machine learning means models which have the same variance distribution across all input data. The "regular" neural network models (shown in Figure 1 (a)) or the linear regression models assume homoskedasticity. Under the assumption of homoskedasticity, the output from a neural network is distributed as:

$$\hat{y} \sim \mathcal{N}(\mu(\boldsymbol{x}), \sigma^2)$$

where the variance σ^2 is assumed constant. Thus, the loss function of the model is given by minimising the negative log-likelihood (equivalent to maximising the likelihood) under the i.i.d. assumption is given as:

$$\text{Loss} = \frac{\sum_{i=1}^{N} (\hat{y}_i - y_i)^2}{N}$$

where y_i and $\hat{y}_i$ denote the ground truth and prediction of the i^{th} (where $i \in \{1 \cdots N\}$) data point. The prediction of the i^{th} data point ($\boldsymbol{x_i}$) can be computed by running the forward pass of the neural network on $\boldsymbol{x_i}$, or, in short, we can write: $\mu_i = NN(\boldsymbol{x_i})$. We can see that minimising the negative log-likelihood naturally leads us to mean squared error as the cost function. While often not discussed, one can calculate the maximum likelihood estimate for the variance σ^2 term as follows:

$$\sigma^2_{MLE} = \frac{1}{N} \sum_{i=1}^{N} (y_i - NN(\boldsymbol{x_i}))^2$$

The loss functions can be optimised using the well-known gradient descent-based approaches. Given the "simplistic" notion of uncertainty from homoskedastic models, in practice, these models are treated as models without uncertainty quantification. Following the usual practice, we do not study the uncertainty estimates from these models in this paper.

2.2 Heteroskedastic Regression

In contrast to the above discussed homoskedastic regression model, heteroskedastic regression model (shown in Figure 1 (b)) can learn different variance for different data points. Thus, in heteroskedastic model, in addition to estimating $\mu(\boldsymbol{x})$ as a function of the input, we also estimate/learn $\sigma(\boldsymbol{x})$ as a function of the input. Similar to homoskedastic regression, we can define the loss (to be minimised) as the negative log-likelihood. But, unlike the homoskedastic regression case, we cannot assume σ^2 to be constant. Thus, can write the loss as follows (ignoring constants):

$$\text{Loss} = \frac{\sum_{i=1}^{N} \left(-\frac{1}{2} \log \sigma^2 - \frac{1}{2\sigma^2} (y_i - \hat{y}_i)^2 \right)}{N} \tag{1}$$

Similar to homoskedastic regression, the loss functions can be optimised using the well-known gradient descent based approaches.

Both the models discussed thus far consider only the data (or aleatoric) uncertainty. They do not consider the uncertainty in estimating the parameters (or the epistemic) uncertainty. One way to obtain epistemic uncertainty is by creating an ensemble of models. We first briefly discuss how we can obtain the prediction and uncertainty from an ensemble.

2.3 Prediction from an ensemble of NNs

We consider an ensemble of two kinds of models: homoskedastic and heteroskedastic models. If we have an ensemble of N homoskedastic NNs and their prediction for an input (obtained via the forward pass) is given as: μ_i, then the predicted mean and standard deviation of the ensemble is calculated by:

$$\mu_{\text{ensemble}} = \frac{\sum_{i=1}^{N} \mu_i}{N} \tag{2}$$

$$\sigma_{\text{ensemble}} = \sqrt{\frac{\sum_{i=1}^{N} (\mu_i - \mu_{\text{ensemble}})^2}{N}} \tag{3}$$

In heteroskedastic regression, for an input datapoint, we predict two neurons that correspond to mean (μ_i) and sigma (σ_i) $\forall i \in \{1, \cdots N\}$, and the output is a Gaussian (or Normal) distribution. The ensemble of N such models will produce a mixture of Gaussian

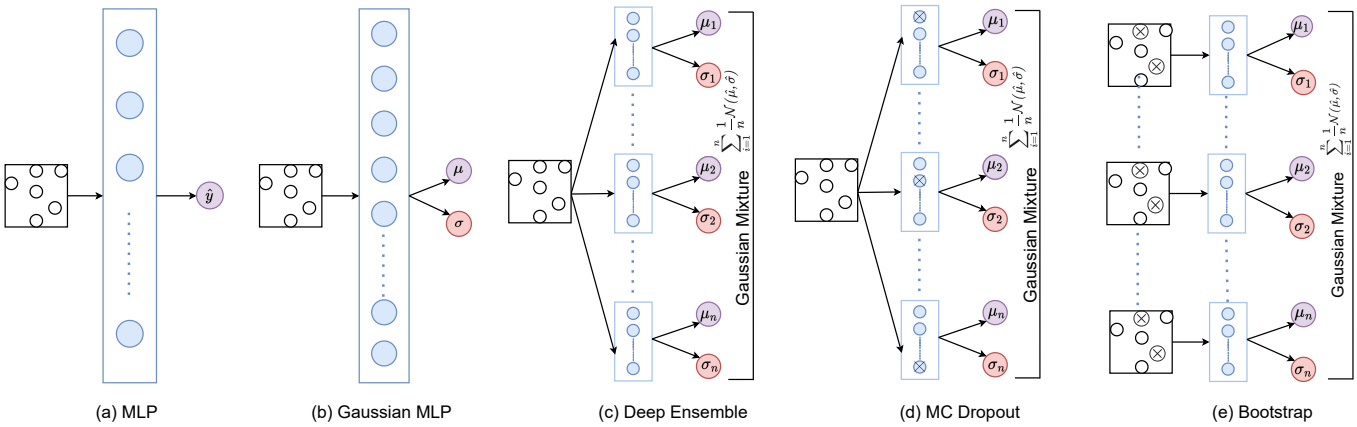


Figure 1: Various neural network model architectures for computing uncertainty in predictions

distributions [17]. Under the assumption that each distribution has an equal weight in the mixture, the resulting mean and sigma are determined. Calculating the predictive mean is same as equation 2. The standard deviation of the ensemble is calculated as:

$$\sigma_{\text{ensemble (Heteroskedastic)}} = \sqrt{\frac{\sum_{i=1}^{N}(\sigma_i^2 + \mu_i^2)}{N} - \mu_{\text{ensemble}}^2} \quad (4)$$

In this paper, we have implemented three ensemble methods, namely Monte Carlo (MC) Dropout, Deep Ensembles and Bootstrap with Homoskedastic and Heteroskedastic models. We now discuss these three methods.

2.4 MC Dropout

Monte Carlo Dropout [8] (called MC Dropout from now onwards) trains a neural network (either homoskedastic or heteroskedastic) as usual. However, at the time of prediction, it randomly drops out nodes from the network. The probability of a node being dropped (or retained) is given as per the Bernoulli distribution. We can note that the operation of MC Dropout is similar to the regular dropout considered in the context of reducing model overfitting [21]. However, the key difference is the application of dropout at test time. Every forward pass of the network can drop out different nodes (given each forward pass accepts a different random seed as an argument), and result in a different prediction for a given input, as shown in Figure 1(d). Importantly, the MC dropout method can be considered as an approximation of Bayesian deep gaussian processes [8]. Thus, the MC dropout method, though simple has strong theoretical properties.

2.5 Bootstrap

Bootstrap aggregating is a technique used to reduce the variance of a machine learning model [4]. It works by training multiple models on different subsets of the data as in Figure 1(e) and then averaging the predictions of all the models. This can help reduce overfitting and improve the overall performance of the model. Using the bootstrap method, unlike the MC dropout method we train N different models independently, each given a different subset of the dataset. Thus, the computation and memory requirement for Bootstrap based method to create an ensemble of NNs is expensive.

2.6 Deep Ensemble

The Deep ensemble method [17] is similar to the bootstrap method and trains N independent models as shown in Figure 1c. The key difference between the bootstrap and deep ensemble model is that each model in the ensemble learns over the entire dataset unlike the bootstrap method The models can be of different types (e.g., different neural network architectures), or they can be different instances of the same type of model (e.g., different random initialisations of the same architecture).

2.7 Quantifying predictive uncertainty

Having discussed various methods of estimating uncertainty using neural networks, we now discuss methods to quantify the "goodness" of the predicted uncertainty (often also called how well a model is calibrated). We explain a well-calibrated model with an example in Figure 2. We take a ground truth or true function $f(x) = x\sin(x)$ and learn a probabilistic model (learning the mean and the standard deviation) over the training data. Then, over the test data, we plot the 90% confidence interval. For the normal distribution, 90% CI refers to the $\mu \pm 1.64\sigma$ band. However, we can see that only 72.5% of the observations fall within the $\mu \pm 1.64\sigma$ band. From here on, we refer to the *chosen* CI as p and the empirically found fraction of points within the band corresponding to the CI of p as $\hat{p}$. Ideally, we would like $\hat{p}$ to be the same as p. In Figure 3, we show the reliability diagram (or calibration curve) for the above-mentioned example. From Figure 3, we can note that the relationship between $\hat{p}$ and p, uncalibrated (shown in blue), lies below the ideal ($\hat{p} = p$) line. It should be noted that to generate such a reliability diagram, we choose varying CIs (p) and then find the corresponding $\hat{p}$.

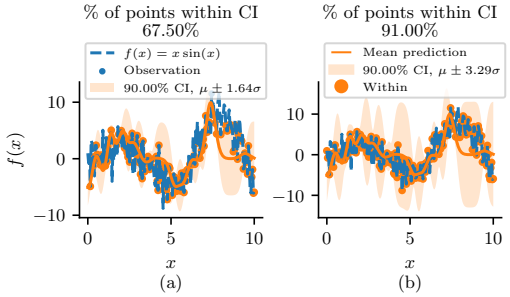


Figure 2: 33% Increase in number of data points lying in 90% Confidence Interval

The reliability curve can help visually understand quality of the model's calibration. We now discuss a quantitative measure called expected calibration error (ECE) to measure the model calibration. To compute the ECE, we choose a set of p values (say, 0.01, 0.02, $\cdots$, 0.99 as an example) and compute the corresponding $\hat{p}(p)$ as a function of p. Finally, we can compute the ECE as:

$$\text{ECE} = \frac{\sum_{p=1}^{P} |\hat{p}(p) - p|}{P}$$

Lower ECE value indicates a better calibration.

2.8 Model Recalibration

We now discuss our proposed method to improve model calibration or to reduce ECE. We continue working with our above running example from Figure 2 and Figure 3. We previously discussed: the *uncalibrated* (shown in blue) relationship between $\hat{p}$ and p lies below the ideal ($\hat{p} = p$) line. To improve this relationship, we want for example 90% CI to correspond to more than the 72.5% points from the uncalibrated model. Therefore, we learn a function g mapping $\hat{p}$ to p using Isotonic regression. Isotonic regression is well-suited for this specific function as it learns monotonically increasing non-parametric relationship. Finally, at test time, say, we want to get 90% of points within the 90% CI, we find $g(90\%)$, which in this example would map to a number higher than 90% (in our example this is: 92.7%) meaning that we need to increase our band in order to capture approximately 90% of the datapoints. Going from CI of 90% on original model to a CI of 92.7% on the original models means increasing the band from $\mu \pm 1.64\sigma$ to $\mu \pm 1.79\sigma$. This in turn, leads to a $\hat{p} = 78\%$, which is greater than the $\hat{p} = 72.5\%$ of the uncalibrated model (Figure 2). Further, we can repeat this procedure for the different values of p to obtain the improved reliability diagram for the calibrated model (shown in orange in Figure 3).

An important detail about the recalibration process is that we fit our NN on the training dataset, recalibrate (or learn the above-mentioned function g) on a previously unseen dataset called the calibration dataset. We split the dataset with 75% training and 25% calibration dataset. It should also be noted that our recalibration procedure only changes the model uncertainty (σ) without affecting the mean (μ) prediction.

3 NEURAL NETWORKS FOR NILM

We now discuss two state-of-the-art NN methods used for NILM.

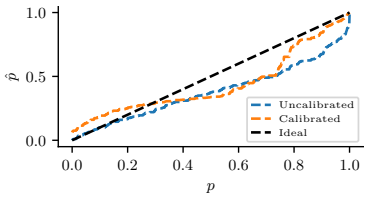


Figure 3: Reliability diagram helps quantify the quality of our uncertainty estimates

Appliance	Training House Number	Testing House Number
Fridge	1, 2, 3, 5	6
Dishwasher	1, 3	2
Microwave	1, 3	2

Table 1: The training and test settings for different appliances for our experiments

3.1 Seq2Point

Seq2Point (S2P) [24] maps a sequence of mains power to a point appliance power prediction. The NN architecture is composed of a sequence of 1d convolution filters and dropout. For more details, we refer the reader to prior research [2, 24].

3.2 BiLSTM with Attention Mechanism

The S2P model while considered the state-of-the-art model, was known to perform poorly on sparsely used appliances such as dishwasher and microwave. Recent work [19] have proposed using bi-directional LSTM models with attention for NILM and have shown these models to work well even for sparsely used appliances. We direct the reader to prior research for the detailed model architecture [20].

4 EVALUATION

We now provide the evaluation setup for answering the questions we raised in the introduction. Our work is fully reproducible. We provide the code in our repository[1].

4.1 Datasets

We have used the publicly available REDD dataset [16] for our research. The dataset consists of several appliances collected over several weeks from six different residences. We used information from three appliances for this study: the refrigerator, dishwasher, and microwave. Other appliances have far less data, and most families do not have access to it. Additionally, the dishwasher and microwave require human operation and are only occasionally utilised, whereas the fridge might be regarded a background appliance that operates without human interaction. Additionally, devices like the dishwasher frequently function in several modes (drying, heating, etc.) and demand varied amounts of power draw for these different states. Furthermore, we downsampled the data for both the mains and the appliances to a minute frequency using the pre-processing routines from NILMTK, as done in prior work [20].

[1] https://github.com/VibhutiBansal-11/NILM_Uncertainty

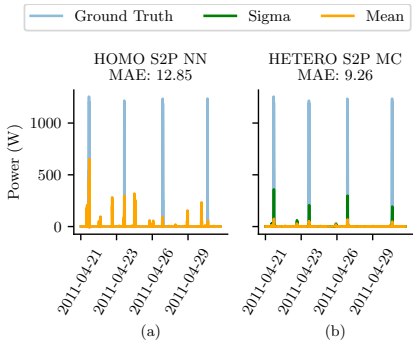


Figure 4: Models incorporating uncertainty such as Heteroskedastic S2P with MC dropout (shown in (b)) can achieve lower MAE than models without uncertainty such as Homoskedastic S2P NN (shown in (a)) for a sparsely used appliance such as the dishwasher. The S2P NN (a) shows high false positives (predicting the dishwasher to be ON when it is actually OFF) in comparison to the Heteroskedastic S2P MC (b).

4.2 Metrics

We use two different metrics to quantify our model performance. First, we use the conventional mean absolute error (MAE) metric defined as following: MAE $= \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|$. Here, n is the number of samples, $\hat{y}_i$ is the predicted appliance reading, and y_i is the ground truth reading of an appliance. The MAE has been used across several prior NILM studies[6]. Further, we note that a lower MAE indicates better performance.

We used expected calibration error (ECE) discussed in Section 2.7 to quantify the uncertainty performance of our model.

4.3 Experimental Setup

We discuss the dataset split chosen for training and testing in Table 1. Our dataset split choice is based on prior literature [20], and on the basis of availability of the appliance data across these homes. In the interest of space, we link the hyperparameter space in a dedicated page in our Github[2]. Like our dataset split, our hyperparameter choices was inspired by previous literature [2, 20]. We used 4 X NVidia A100 GPUs for training our models, and JAX[3] and Flax[4] for creating our neural network models. All our models are compatible in the NILMTK ecosystem [1, 2].

4.4 Results and Analysis

We now present our results based on the questions we raised in the introduction section of this paper. The main result in Table 2 compares the MAE and the ECE across the different models and appliances.

[2]https://github.com/VibhutiBansal-11/NILM_Uncertainty/blob/master/hyperparameters.md

[3]https://jax.readthedocs.io

[4]https://flax.readthedocs.io/en/latest/overview.html

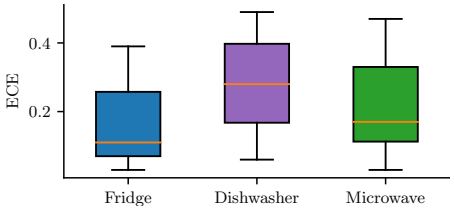


Figure 5: The expected calibration error (ECE) quantifying calibration performance for three appliances across different models presented in Table 2 is generally lower for the fridge compared to sparsely used appliances like the dishwasher and the microwave.

4.4.1 *Do NNs with uncertainty achieve comparable error on conventional metrics to the baselines?*

We can see from Table 2, that the MAE for the models that provide uncertainty such as S2P homoskedastic (MC, DE, BS), and heteroskedastic (NN, MC, DE, BS), is comparable or better than the baseline (Homoskedastic S2P/LSTM) models that do not incorporate uncertainty. As an example, the S2P homoskedastic model has a MAE of 26.16 for the fridge, whereas S2P homoskedastic with DE achieves a lower MAE of 24.73. Similarly, for the LSTM-based models, the MAE among model variants with uncertainty is comparable to the MAE of models without uncertainty. We believe that our finding that **we can achieve comparable or better performance on conventional metrics while additionally incorporating the notion of uncertainty** is an important finding for the community going forward.

Interestingly, we can significantly reduce the MAE for the dishwasher by using models incorporating uncertainty. For example, the S2P homoskedastic NN has a MAE of 12.85 compared to the improved MAE of 9.26 for the S2P heteroskedastic model with MC Dropout. We now explain this finding in Fig 4, where we can see that baselines S2P homoskedastic NN (MAE of 12.85) is better at predicting the peaks (ground truth) but it is also giving high false positives (wrongly predicting the dishwasher to be ON when it is actually OFF) which increases its MAE. While the model incorporating uncertainty (S2P heteroskedastic MC Dropout with MAE error of 9.26) does not predict the peaks well but has less false positives which results in lower MAE. Further, from Fig 4(b), we can also observe the uncertainty in the prediction (sigma, shown in green). is higher when the dishwasher changes state from OFF to ON. **Higher uncertainty when an appliance changes state is expected as the model is likely to be uncertain during the transition and will likely get more confident once it observes more samples from the changed state**.

4.4.2 *Are certain appliances or appliance states more prone to poor calibration?*

From Fig 5 and Table 2, we can observe that the ECE for the fridge is generally lower than that of sparsely used appliances (dishwasher and microwave). As discussed earlier, and in prior literature [20], NILM methods generally perform worse for sparsely used appliances in comparison to appliances such the fridge or air conditioner.

	Fridge			Dishwasher			Microwave		
	MAE	ECE	C.ECE	MAE	ECE	C.ECE	MAE	ECE	C.ECE
Model : S2P									
Homoskedastic									
NN	26.16	-	-	12.85	-	-	11.18	-	-
MC	26.22	0.25	0.23	12.97	0.48	0.42	11.17	0.47	0.45
DE	24.73	0.26	0.27	12.46	0.43	0.37	**11.16**	0.43	0.35
BS	**24.69**	0.24	0.26	11.49	**0.06**	0.21	11.25	0.30	0.18
Heteroskedastic									
NN	26.91	0.13	0.05	9.61	0.19	**0.06**	12.46	**0.03**	0.08
MC	26.38	**0.03**	0.19	**9.26**	0.16	0.12	12.56	0.05	0.18
DE	26.85	0.04	0.03	9.81	0.13	0.06	12.66	0.20	**0.06**
BS	26.68	0.07	0.11	10.21	0.43	*0.17*	13.27	0.22	0.35
Model: LSTM									
Homoskedastic									
NN	36.51	-	-	12.29	-	-	16.76	-	-
MC	36.57	0.33	*0.19*	12.29	0.49	0.39	16.77	0.43	*0.25*
DE	37.05	0.39	*0.25*	10.33	0.29	0.21	15.60	0.34	0.17
BS	36.66	0.37	*0.23*	9.99	0.14	0.18	14.80	0.12	0.15
Heteroskedastic									
NN	32.80	0.06	0.07	10.07	0.23	0.07	12.61	0.12	0.14
MC	32.83	0.07	0.04	10.11	0.27	0.05	12.61	0.14	0.12
DE	33.38	0.08	0.06	10.07	0.30	0.05	12.81	0.11	0.13
BS	33.16	0.09	**0.02**	11.12	0.29	0.07	13.06	0.04	0.11

Table 2: Mean absolute error (MAE), Expected Calibration Error pre calibration (ECE) and Expected Calibration Error post calibration (C. ECE) for three appliances across 16 model variants. The best performing model for each metric has been made bold and the value of C.ECE has been made italic where the improvement of error (C.ECE - ECE) is maximum

However, our findings suggest that not only is the performance in terms of MAE poor for sparsely used appliances, but, the models with uncertainty also have worse calibration for sparsely used appliances, in comparison to regularly used appliances Thus, these models present significant scope in improving both the MAE as well as calibration performance.

We now discuss the reliability diagram and the predicted power for the three appliances across a subset of the models. We choose the models and the plotted time window for illustrative purposes. However, it should be noted that the reported computed metrics are for the entire dataset as shown in Table 2. We first discuss the results for fridge. In Figure 6(a) we observe that the prediction for the Homoskedastic S2P model with bootstrap matches the ground truth well, resulting in a low MAE of 24.69. However, the ECE of 0.25 is high in comparison to other models (as seen from Figure 6(d)). From the reliability diagram (for now we direct the reader to only study the curve labelled *Total*) we can observe that the empirical fraction of points ($\hat{p}$) is below the ideal line ($\hat{p} = p$ line). This indicates that the learnt model is over-confident, i.e. the model thinks that it predicts the mean well and thus needs a low uncertainty band. However, if the uncertainty of the model were increased, especially during the fridge ON states (around 07:30 to 08:10 hours, 08:40 to 09:30 hours, and 10:10 to 10:40 hours), the calibration performance will improve (reduction in ECE).

From Figure 6(b), we can note that the prediction during the ON state is wrong. However, interestingly, the uncertainty in the prediction (sigma, shown in green) is high, especially during the time when the prediction is particularly bad. The high uncertainty helps in achieving a well-calibrated model. We confirm this in Figure 6(e), where we observe that the empirical fraction of points (for *Total* curve) ($\hat{p}$) is close to the ideal line ($\hat{p} = p$ line). From Figure 6(c), we can observe the predictions for Heteroskedastic LSTM model are comparable in terms of ECE to Heteroskedastic S2P model with bootstrap shown in Figure 6(b). However, interestingly, the corresponding calibration curves (Figure 6(e) and (f)) are substantially different when we consider the calibration curves separately for the ON and the OFF states. In Figure 6(e), the model is under-confident, i.e. it predicts a high value of uncertainty for both the ON and the

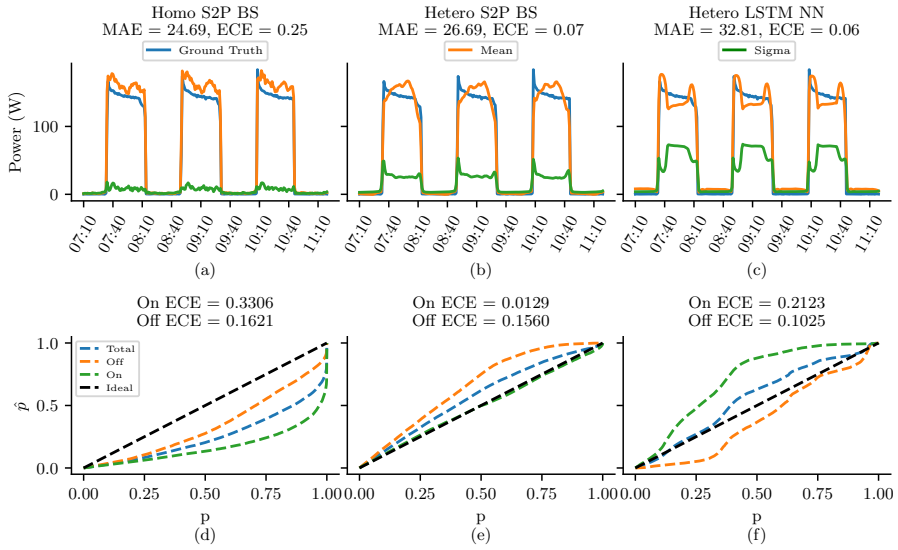


Figure 6: Predicted power and reliability diagrams for fridge across different models (a) Homoskedastic S2P model with bootstrap showing the best MAE (lowest) (b) Heteroskedastic S2P model with bootstrap showing a comparitively low MAE but low ECE (c) Heteroskedastic LSTM model showing a high MAE but low ECE

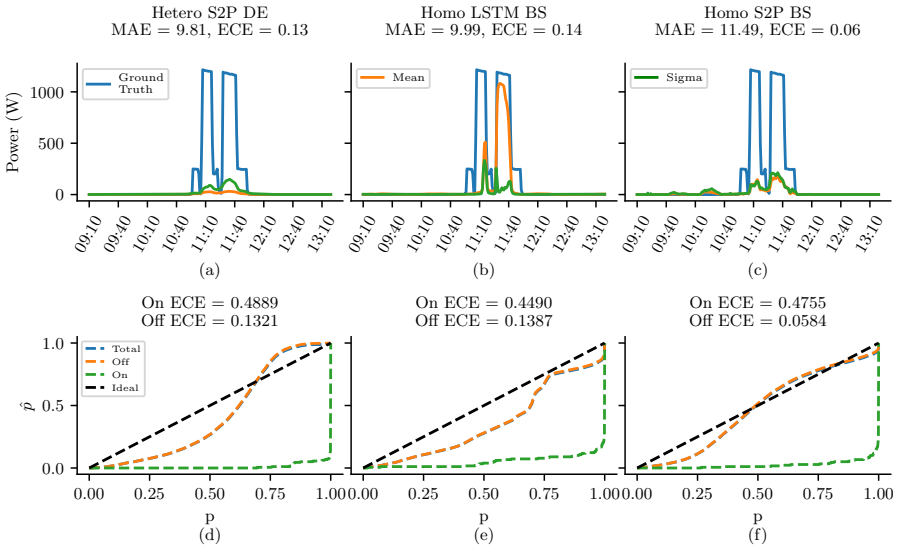


Figure 7: Predicted power and reliability diagrams for dish washer across different models (a) Heteroskedastic S2P model with deep ensemble showing the low MAE (b) Homoskedastic LSTM model with bootstrap showing a high MAE and high ECE but low ECE for ON-state (c) Homoskedastic S2P model with bootstrap showing a high MAE but best ECE (lowest)

OFF states. However, in Figure 6(f), the model is over-confident for the OFF state and under-confident for the ON state. These findings highlight that **different appliance states can have highly varying calibration curves**, and **we can achieve an overall low calibration error if the individual states calibration errors cancel out each other**.

We now discuss the calibration and predicted power for dishwasher. The homoskedastic S2P BS (Figure 7(c) and (f)) ECE value is low, corresponding to 0.06 but we are unable to obtain an uncertainty estimate that can capture the ON state ground truth within an appropriate confidence interval because the model is overconfident in this state. However, as the data is largely biased towards the OFF state, the final ECE values are low and the *Total* $\hat{p}$ is close to the $\hat{p} = p$ line. Similarly, for the heteroskedastic S2P DE (Figure 7(a) and (d)). In contrast, in Figure 7(b) for the homoskedastic LSTM with bootstrap model, the ECE is comparable (0.14) to homoskedastic

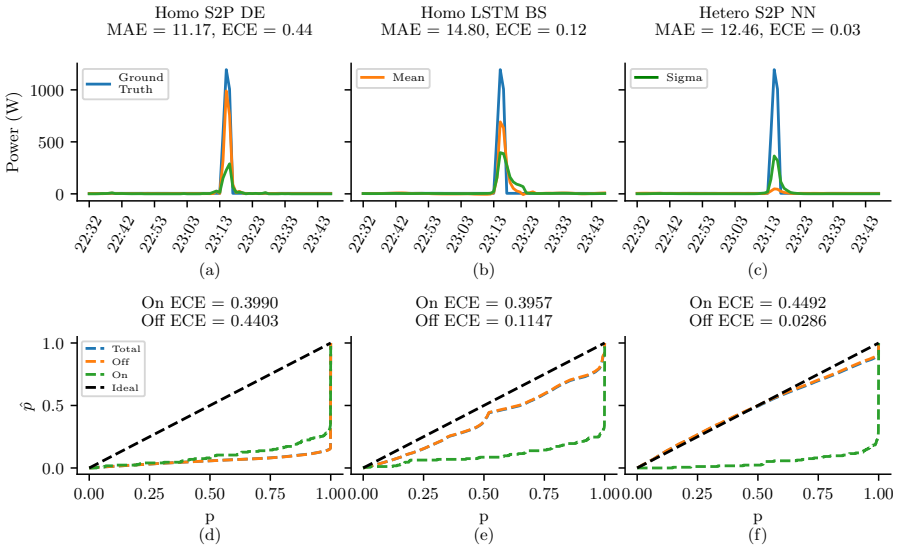


Figure 8: Predicted power and reliability diagrams for microwave across different models (a) Homoskedastic S2P model with deep ensemble showing the best MAE (lowest) (b) homoskedastic LSTM model with bootstrap showing a high MAE but low ECE (trade-off) (c) Heteroskedastic S2P model showing a high MAE but best ECE (lowest)

S2P BS (Figure 7(c)). However, the model is doing a better job at predicting the second peak (around 11:10 to 11:40 hours). Furthermore, the uncertainty (sigma, shown in green) increases on the right side of the second peak (around 11:40 hours), where the predicted power deviates from the ground truth. We confirm from Figure 7(e), that indeed for homoskedastic LSTM model with bootstrap, the ECE for the ON state is better than the other compared models. This leads us to our next learning: **A good ECE may hide the imbalance between the different states, and for a thorough analysis, it is recommended to consider the state-wise ECE.**

Our findings on similar experiments done on the microwave (Figure 8) are comparable to the findings for fridge and dishwasher. Importantly, as expected, **estimating the uncertainty accurately (low ECE) for the ON state for sparse appliances like dishwasher and microwave is non-trivial**. Overall, from the above experiments, we can conclude that **there is an important trade-off between the two considered metrics: MAE and ECE, and different applications may call for a nuanced choice of metric for evaluating performance.**

4.4.3 *Can recalibration improve model uncertainty?*

We applied our isotonic regression-based recalibration approach previously discussed in Section 2.7. From Table 2, we note that the ECE post calibration (C.ECE) improves (reduces) for most models in comparison to the ECE before calibration. We now dive deeper into some specific illustrative examples to show the effect of recalibration on the three appliances. We use the 95% confidence interval (CI) for our experiments. First, in Figure 9(a) we can observe that S2P homoskedastic MC model originally had 65% of points ($\hat{p} = 0.65$) in 95% confidence interval ($p = 0.95$) which is visibly improved after recalibration to 80% ($\hat{p} = 0.80$) in Figure 9(b). We can further confirm from Figure 9(c) that the reliability diagram improves post-calibration. Importantly, we can note from Figure 9(b) that a much higher proportion of the observation during the ON state (07:30 to 08:10) now fall within the CI, in comparison to Figure 9(a).

Similarly, for the dishwasher (Figure 10) and microwave (Figure 11) there is an improvement in model uncertainty as quantified by the reduction of the gap between p and $\hat{p}$. We may also note from Figure 10 and Figure 11, that the uncertainty quantification improves for both the ON and the OFF states. However, quantifying uncertainty for the ON state even post calibration has a significant scope for improvement.

We now analyse why ECE can increase post calibration for some models and appliances as shown in Table 2. This trend can be attributed to the contrasting nature of confidence of the calibration set and test set. We can see from Figure 12(a) that 90% of points lie in 95% CI before calibration for the fridge for heteroskedastic S2P MC dropout model. Instead of increasing from 90 to 95% (ideally) post calibration, we observe only 84% points which is worse and hence increases the test ECE as seen in Figure 13(b). The calibration set curve before calibration was under confident (above the ideal line), as seen in Figure 13. Thus, to match the ideal curve, post calibration, the $\hat{p}$ would be reduced for the same value of p. On the contrary, the model before calibration was overconfident on the test set, where $\hat{p}$ was below the ideal curve. Thus, recalibration further pushes down $\hat{p}$ making the ECE worse. We can thus conclude that **good recalibration requires similar characteristics between the calibration and the test set.**

5 LIMITATIONS, DISCUSSION AND FUTURE WORK

(1) In the future we plan to study performance of the 14 model variants on more datasets and appliances.
(2) In this paper we assumed normal distribution (as is the standard in the machine learning community). In the future, we

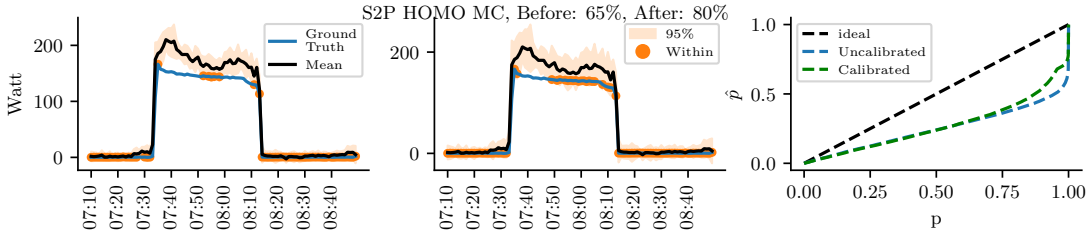


Figure 9: Effect of recalibration on fridge for homoskedastic S2P MC dropout model: (a) Corresponding to the 95% confidence interval, our uncalibrated model has only 65% of the observed data points; (b) the calibrated model in contrast has a higher fraction of 80% points; (c) the reliability diagram showing the improvement in uncertainty quantification post calibration.

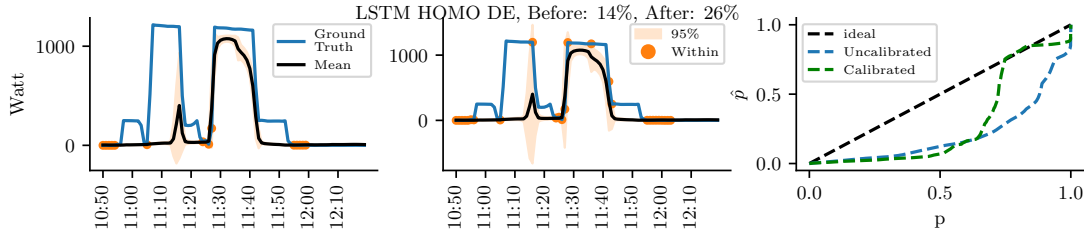


Figure 10: Effect of recalibration on dishwasher for homoskedastic LSTM Deep Ensemble Model: (a) Corresponding to the 95% confidence interval, our uncalibrated model has only 14% of the observed data points; (b) the calibrated model in contrast has a higher fraction of 26% points; (c) the reliability diagram showing the improvement in uncertainty quantification post calibration

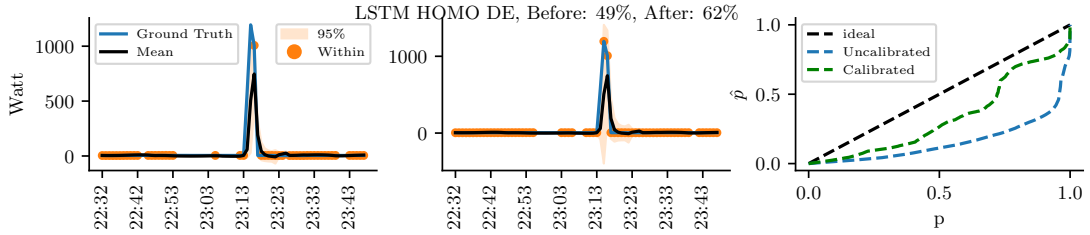


Figure 11: Effect of recalibration on microwave for homoskedastic LSTM deep ensemble model: (a) Corresponding to the 95% confidence interval, our uncalibrated model has only 49% of the observed data points; (b) the calibrated model in contrast has a higher fraction of 62% points; (c) the reliability diagram showing the improvement in uncertainty quantification post calibration

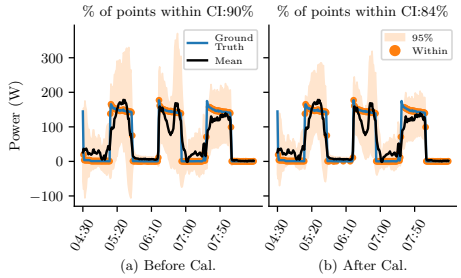
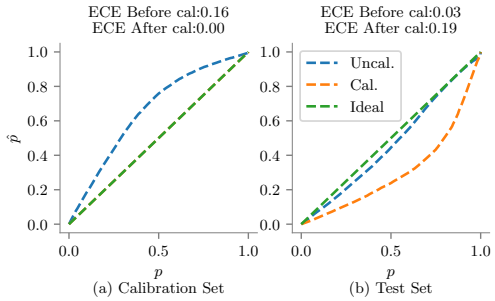


Figure 12: Effect of recalibration for fridge for heteroskedastic S2P model with MC Dropout - (a) Before Calibration 90% points in 95% CI (b) After Calibration 84% points in 95% CI

plan to study likelihoods such as log-normal distribution to strictly enforce non-negativity.

(3) In this paper we presented several approximate inference methods like Deep Ensemble, MC Dropout and Bootstrapping to obtain uncertainty. While we also evaluated on Markov Chain Monte Carlo (MCMC) based methods (where methods like NUTS [11]) on NILM data, we did not report the results as the experiments are significantly more time consuming. However, in the future, we plan to compare our methods proposed in this paper to MCMC based methods too.

(4) We discussed that the different characteristics of the calibration and test set can result in recalibration making the uncertainty performance worse. In the future, we plan to study techniques to understand the suitability of a given



(a) Calibration Set (b) Test Set

Figure 13: Test ECE increases by 0.13 after calibration for fridge for heteroskedastic S2P model with MC Dropout (a) Model is underconfident on calibration set, it becomes ideally confident with 0 ECE after recalibration (b) Model was overconfident before calibration and confidence after recalibration increases even more due to the nature of calibration set making ECE worse

calibration set to improve the uncertainty performance on an unseen test dataset.

(5) In NILM applications, the data distribution can drift over time owing to reasons such as: i) changes in weather conditions; ii) appliance wear and tear; iii) change in operational usage; etc. OOD examples are unlikely to contain the same patterns as training distribution examples. This may limit the generalization ability. Thus, in the future, we plan to study the uncertainty quantification for the out of distribution (O.O.D.) setting.

6 CONCLUSIONS

NN methods have proven to be the state-of-the-art models for NILM. In this paper, we have shown how to adapt existing architectures to provide predictive uncertainty. We took a NILM specific flavour to our work and discuss our findings in the NILM context. As an example, we showed that calibration needs to be studied separately for different states of an appliance. Finally, we have highlighted the shortcomings of existing approaches to quantify uncertainty. We hope this paper opens discussions in the NILM and the BuildSys community around predictive uncertainty.

7 ACKNOWLEDGEMENTS

We would like to thank Cisco Research for sponsoring this research. For this project, Vibhuti Bansal was supported via the Google exploreCS Research 2022 fellowship.

REFERENCES

[1] Nipun Batra, Jack Kelly, Oliver Parson, Haimonti Dutta, William Knottenbelt, Alex Rogers, Amarjeet Singh, and Mani Srivastava. 2014. NILMTK: an open source toolkit for non-intrusive load monitoring. In *Proceedings of the 5th international conference on Future energy systems*. 265–276.

[2] Nipun Batra, Rithwik Kukunuri, Ayush Pandey, Raktim Malakar, Rajat Kumar, Odysseas Krystalakos, Mingjun Zhong, Paulo Meira, and Oliver Parson. 2019. Towards reproducible state-of-the-art energy disaggregation. In *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 193–202.

[3] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. 2015. Weight Uncertainty in Neural Network. In *Proceedings of the 32nd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 37)*, Francis Bach and David Blei (Eds.). PMLR, Lille, France, 1613–1622. https://proceedings.mlr.press/v37/blundell15.html

[4] Leo Breiman. 1996. Bagging predictors. *Machine learning* 24, 2 (1996), 123–140.

[5] Sarah Darby et al. 2006. The effectiveness of feedback on energy consumption. *A Review for DEFRA of the Literature on Metering, Billing and direct Displays* 486, 2006 (2006), 26.

[6] Anthony Faustine, Lucas Pereira, Hafsa Bousbiat, and Shridhar Kulkarni. 2020. UNet-NILM: A Deep Neural Network for Multi-Tasks Appliances State Detection and Power Estimation in NILM. In *Proceedings of the 5th International Workshop on Non-Intrusive Load Monitoring* (Virtual Event, Japan) *(NILM'20)*. Association for Computing Machinery, New York, NY, USA, 84–88. https://doi.org/10.1145/3427771.3427859

[7] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of The 33rd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 48)*, Maria Florina Balcan and Kilian Q. Weinberger (Eds.). PMLR, New York, New York, USA, 1050–1059. https://proceedings.mlr.press/v48/gal16.html

[8] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning *(ICML'16)*. JMLR.org, 1050–1059.

[9] George William Hart. 1992. Nonintrusive appliance load monitoring. *Proc. IEEE* 80, 12 (1992), 1870–1891.

[10] Kanghang He, Lina Stankovic, Jing Liao, and Vladimir Stankovic. 2016. Non-intrusive load disaggregation using graph signal processing. *IEEE Transactions on Smart Grid* 9, 3 (2016), 1739–1747.

[11] Matthew D. Hoffman and Andrew Gelman. 2011. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. https://doi.org/10.48550/ARXIV.1111.4246

[12] Patrick Huber, Alberto Calatroni, Andreas Rumsch, and Andrew Paice. 2021. Review on deep neural networks applied to low-frequency nilm. *Energies* 14, 9 (2021), 2390.

[13] Jack Kelly and William Knottenbelt. 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*. 55–64.

[14] J Zico Kolter, Siddharth Batra, and Andrew Y Ng. 2010. Energy disaggregation via discriminative sparse coding. In *Advances in Neural Information Processing Systems*. 1153–1161.

[15] J Zico Kolter and Tommi Jaakkola. 2012. Approximate inference in additive factorial hmms with application to energy disaggregation. In *Artificial intelligence and statistics*. 1472–1482.

[16] J. Zico Kolter and Matthew J. Johnson. 2011. REDD: A Public Data Set for Energy Disaggregation Research. In *IN SUSTKDD*.

[17] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems* 30 (2017).

[18] Max-Heinrich Laves, Sontje Ihler, Jacob F. Fast, Lüder A. Kahrs, and Tobias Ortmaier. 2020. Well-Calibrated Regression Uncertainty in Medical Imaging with Deep Learning. In *Proceedings of the Third Conference on Medical Imaging with Deep Learning (Proceedings of Machine Learning Research, Vol. 121)*, Tal Arbel, Ismail Ben Ayed, Marleen de Bruijne, Maxime Descoteaux, Herve Lombaert, and Christopher Pal (Eds.). PMLR, 393–412. https://proceedings.mlr.press/v121/laves20a.html

[19] Veronica Piccialli and Antonio M. Sudoso. 2021. Improving Non-Intrusive Load Disaggregation through an Attention-Based Deep Neural Network. *Energies* 14, 4 (Feb 2021), 847. https://doi.org/10.3390/en14040847

[20] Hetvi Shastri and Nipun Batra. 2021. Neural Network Approaches and Dataset Parser for NILM Toolkit. In *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation* (Coimbra, Portugal) *(BuildSys '21)*. Association for Computing Machinery, New York, NY, USA, 172–175. https://doi.org/10.1145/3486611.3486652

[21] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15 (2014), 1929–1958.

[22] Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. 2020. Uncertainty Estimation Using a Single Deep Deterministic Neural Network. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 9690–9700. https://proceedings.mlr.press/v119/van-amersfoort20a.html

[23] Yanwu Yang, Xutao Guo, Yiwei Pan, Pengcheng Shi, Haiyan Lv, and Ting Ma. 2021. Uncertainty Quantification in Medical Image Segmentation with Multi-decoder U-Net. https://doi.org/10.48550/ARXIV.2109.07045

[24] Chaoyun Zhang, Mingjun Zhong, Zongzuo Wang, Nigel Goddard, and Charles Sutton. 2018. Sequence-to-point learning with neural networks for non-intrusive load monitoring. In *Thirty-second AAAI conference on artificial intelligence*.